

Adaptive Multistage Stochastic Programming

Citation for published version (APA):

Kayacık , S. E., Basciftci, B., Schrottenboer, A., & Ursavas, E. (2024). *Adaptive Multistage Stochastic Programming*. arXiv.org. <https://arxiv.org/abs/2401.07701>

Document license:

CC BY

Document status and date:

Published: 15/01/2024

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Adaptive Multistage Stochastic Programming

Sezen Ece Kayacik

Department of Operations, Faculty of Economics and Business, University of Groningen, the Netherlands, s.e.kayacik@rug.nl

Beste Basciftci

Department of Business Analytics, Tippie College of Business, University of Iowa, Iowa City, Iowa, the United States,
beste-basciftci@uiowa.edu

Albert H. Schrotenboer

Operations, Planning, Accounting & Control Group, School of Industrial Engineering, Eindhoven University of Technology,
the Netherlands, a.h.schrotenboer@tue.nl

Evrin Ursavas

Department of Operations, Faculty of Economics and Business, University of Groningen, the Netherlands, e.ursavas@rug.nl

Multistage stochastic programming is a powerful tool allowing decision-makers to revise their decisions at each stage based on the realized uncertainty. However, in practice, organizations are not able to be fully flexible, as decisions cannot be revised too frequently due to their high organizational impact. Consequently, decision commitment becomes crucial to ensure that initially made decisions remain unchanged for a certain period. This paper introduces adaptive multistage stochastic programming, a new optimization paradigm that strikes an optimal balance between decision flexibility and commitment by determining the best stages to revise decisions depending on the allowed level of flexibility. We introduce a novel mathematical formulation and theoretical properties eliminating certain constraint sets. Furthermore, we develop a decomposition method that effectively handles mixed-integer adaptive multistage programs by adapting the integer L-shaped method and Benders decomposition. Computational experiments on stochastic lot-sizing and generation expansion planning problems show substantial advantages attained through optimal selections of revision times when flexibility is limited, while demonstrating computational efficiency of the proposed properties and solution methodology. Optimizing revision times in a less flexible case can outperform arbitrary selection in a more flexible case. By adhering to these optimal revision times, organizations can achieve performance levels comparable to fully flexible settings.

Key words: Stochastic Programming, Multistage Stochastic Optimization, Mixed-Integer Programming, Lot Sizing, Generation Expansion Planning

1. Introduction

Organizations frequently need to make decisions with lasting impact while facing an uncertain future. This is complex, in particular, when such decisions are made sequentially because every decision taken ‘now’ should anticipate the impact of future information and decisions (Powell 2019). Theoretically, high-quality decisions require *decision flexibility*: the ability to revise earlier made decisions to act upon the latest information, which within a business context translates into ‘agility’ (Mankins 2022b). However, many organizations cannot adopt agile practices because many decisions cannot be revised too frequently due to their high organizational impact. This necessitates *decision commitment* to assure that initially made decisions remain unchanged for a certain

amount of time despite new information or evolving circumstances. In practice, decision-makers must balance decision flexibility with decision commitment; they limit the number of revisions and identify *when* to revise in advance of the actual revision decision. The right balance will result in revising decisions only at the most urgent times, while otherwise committing to decisions previously made. In this study, we provide the first general stochastic optimization approach that determines the optimal revision times to balance flexibility and commitment in stochastic decision-making.

Decision flexibility plays a significant role in effectively addressing challenges in today's uncertain business environment (Gallo 2010, Mankins 2022a). However, it is crucial to acknowledge that frequently revising an earlier-made decision (based on new information) is not always favorable or feasible in real life. This is acknowledged in supply chain management (Wadhwa et al. 2008), economics (Rudloff et al. 2014), and policy-making (Chemama et al. 2019). For example, in supply chain management, producers often have constraints that limit frequently adjusting production quantities and schedules. These limitations can arise from various factors such as production capacity, high setup costs, the need to allocate sources among multiple customers, and predetermined production schedules for other products that cannot be easily changed (Kouvelis et al. 2006, Milner and Kouvelis 2005). Besides, a recent survey by Aminov et al. (2019) highlights the time-consuming nature of decision-making, revealing that respondents spend over 30% of their working hours on this task, much of which is used ineffectively. For such time-consuming decision revisions, businesses may opt for limiting the number of revisions. In other words, finding the right balance between decision flexibility and commitment is essential.

As another relevant area, the challenges faced in the energy sector, specifically through the transition to climate-neutral energy systems, necessitate a balance between decision flexibility and commitment. Long-term energy policy decisions must effectively incorporate the inherent uncertainties of the energy system, technological developments, and societal changes (Köhler et al. 2019). Meadowcroft (2009) emphasizes the importance of flexibility and revising decisions based on new information while, at the same time, ensuring reliability to investors and stakeholders during long-term energy transitions. Accordingly, policymakers should announce their policies and potential revision timelines years in advance. This, for example, is required in developing long-term transition plans toward a green hydrogen economy in the Northern Netherlands, where Europe's first hydrogen valley is being built (New Energy Coalition 2020). In such strategic plans, which involve costly investment decisions and multiple stakeholders, frequent changes are undesirable due to the potential disruptions they may cause. Additionally, contractual limitations or the lengthy process required to establish the necessary infrastructure for the energy system can limit flexibility (Munoz et al. 2013, Careri et al. 2011), making commitments requisite during planning.

Latorre et al. (2003) emphasize the need for theory and tools that consider the limited flexibility level of energy expansion planning process to adapt to uncertainties in energy markets.

While ignoring the need for decision commitment, many optimization paradigms allow for *full* decision flexibility. Multistage stochastic programming is, arguably, one of most popular approaches amidst such optimization paradigms (Birge 1985, Daryalal et al. 2022, Bertsimas et al. 2023); it facilitates sequential decision-making under uncertainty by revising decisions throughout a multistage planning horizon based on the latest information. Although this approach has been widely applied in different application domains (e.g., Gul et al. (2015), Pan and Guan (2016)), its widespread adoption in practice can be challenging for various problems due to the limited flexibility required within the decision-making processes, preventing revising decisions at each stage. To address this challenge, this paper advances multistage stochastic programming by incorporating the need for limited flexibility and commitments through optimally determining the most critical stages *when* decisions can be revised.

We introduce a new optimization approach called *adaptive multistage stochastic programming*, which limits the number of revisions allowed throughout the planning horizon for a subset of decisions that require a certain level of commitment. In the first stage, we determine the optimal stages of revisions for these decisions. Subsequently, at these selected stages, these decisions can be revised similarly to classic multistage stochastic programming, while in the remaining stages, commitment is ensured. The clear advantage of our approach is that at the start of the planning horizon, it is already known when decisions will be revised, which allows for effective communication between dependent actors in decision-making contexts. In the remainder of this section, we discuss related work and summarize our contributions by introducing adaptive multistage stochastic programming.

1.1. Related Work

Stochastic programming is one of the most effective analytical techniques for decision-making under uncertainty (Birge and Louveaux 2000). It involves modeling the stochastic process using scenario trees (Shapiro et al. 2009), which represent the possible uncertainty realizations at different stages. There are two fundamental techniques in stochastic programming: two-stage and multistage, which differ in how they handle sequential decision-making under uncertainty. These techniques generally involve two types of decisions: “here-and-now” decisions that can be immediately executed, and “wait-and-see” decisions that depend on the system’s state (i.e., the observation of the system in each decision stage), influenced by prior decisions and observations of uncertain parameters. While two-stage stochastic programming provides fixed decisions upfront planning for here-and-now decisions, multistage stochastic programming extends the framework by revising all decisions at each stage based on the uncertainty realized so far (Birge 1985).

One critical type of uncertainty that arises in stochastic optimization is endogenous uncertainty, which originates from the decisions of the decision-maker. Endogenous uncertainty adds an additional layer of complexity to the associated optimization problems, as the decision-maker must consider not only the uncertain conditions they face but also the potential impact of their own decisions on the underlying stochastic process in future decision stages. Most often, the stochastic programming literature considers exogenous uncertainty, where decisions do not impact the underlying stochastic process. Relatively limited attention has been paid to endogenous uncertainty. In this type of uncertainty, decisions can change the uncertainty realizations by affecting the underlying probability distributions (see, e.g., Hellemo et al. 2018, Basciftci et al. 2021, Şafak et al. 2022, Drusvyatskiy and Xiao 2022) or alternatively the time which uncertainty realization is observed (see, e.g., Goel and Grossmann 2006, Gupta and Grossmann 2011, Apap and Grossmann 2017), where the latter setting will be discussed further.

The endogenous uncertainty literature involving this second line of research allows revising decisions in each stage based on the observed uncertainty. However, it is important to note that uncertainty is not automatically realized in each stage, and its realization depends on the specific decisions made. Until these decisions are made, it remains unknown which uncertain parameters will be observed, and there is a possibility that some uncertain parameters may not be observed at all. To address this, non-anticipativity constraints (NACs) are commonly used for restricting the feasible set of decision variables depending on the time of observed uncertainty, where the number of NACs exponentially increases with the number of possible uncertainty realizations. Goel and Grossmann (2006) present a mixed-integer disjunctive formulation for stochastic programming that enforces endogenous uncertainty with conditional NACs and investigates theoretical properties to eliminate a particular set of NACs. Despite the elimination techniques, solving the reduced models becomes challenging as the model size increases. For example, given that a small number of NACs are active at the optimal solution, Colvin and Maravelias (2010) develop a branch and cut algorithm in which a certain set of NACs are initially removed and reinserted if they are violated within the search tree. In a subsequent study, Gupta and Grossmann (2011) propose NACs elimination together with heuristic solution strategies: including NACs only up to a limited number of stages, and using Lagrangean relaxation of the NACs to decompose the problem into scenarios, which is later extended to the settings with both endogenous and exogenous uncertainties (Apap and Grossmann 2017). To reduce the number of NACs, Boland et al. (2016) provide general conditions depending on the characteristics of the scenario structure. As another relevant line of research, Gupta et al. (2016) and Singla (2018) study stochastic probing algorithms for specific problems, determining uncertainty realization through probing decisions and associated costs. Although those studies show similarities as the decisions affect future conditions and their setting

requires NACs, they differ from our study by making the uncertainty realization conditional and consequently not fully realizing the underlying stochastic process. Furthermore, existing studies do not focus on providing a generic optimization under uncertainty framework under partial flexibility, and present different formulation techniques and NACs reductions that are not directly applicable to our proposed setting.

This paper introduces Adaptive Multistage Stochastic Programming; a new optimization approach that differs from the existing literature by leveraging decision commitment into multistage stochastic programming. It determines at the start of the planning horizon *when* revisions can be made subject to a limited number of allowed revisions. Basciftci et al. (2019) propose an approach called adaptive two-stage stochastic programming in which they optimize the timing of a single decision revision. They propose analytical approaches over a capacity expansion planning problem through a bound analysis of the adaptive two-stage stochastic program in comparison to two-stage and multistage stochastic programming alternatives, based on the timing of the revision decision. Although this is the first paper to consider revisions as decision variables, we observe two restrictions in their model: i) their model allows revising the decision only once, which is restrictive if the planning horizon becomes larger, and ii) they do not provide elimination techniques for the NACs that would reduce the computational burden of the problem. Besides, their study does not provide a generic solution algorithm, but rather approximation algorithms tailored to address the capacity expansion planning problem.

Even without the NACs needed for Adaptive Multistage Stochastic Programming, solving a classic multistage stochastic programming problem is theoretically and computationally challenging. Furthermore, as the number of stages increases, the size of the scenario tree grows exponentially, making it even more difficult to solve (Shapiro and Nemirovski 2005). Decomposition methods are commonly used to solve large-scale optimization models, and two well-known approaches are the integer L-shaped method (Laporte and Louveaux 1993, Angulo et al. 2016) and Benders decomposition (Angulo et al. 2016, Fischetti et al. 2017, Rahmaniani et al. 2020), which can be further utilized over stochastic programs. The L-shaped method is a cutting plane technique that decomposes a model into master problem and subproblems to separate complicating variables from the remainder of the model (Van Slyke and Wets 1969). It was originally developed for continuous subproblems and later improved to the integer L-shaped method (Laporte and Louveaux 1993), which handles pure binary decisions in the master problem and mixed-integer decisions in the subproblem. Benders decomposition is another approach that solves models with mixed-integer master problems and linear subproblems (Benders 1962). We provide a solution approach for generic adaptive multistage stochastic programs with mixed-integer variables by adapting these decomposition methods and leveraging the structure of our proposed formulation.

1.2. Contributions

Summarizing, we make the following scientific contributions.

- We introduce adaptive multistage stochastic programming: a novel and generic decision-making approach that incorporates a balance between decision flexibility and commitment into multistage stochastic programming. By considering the flexibility level of the decision-making process, we set a maximum limit on the number of revisions. The main factor in our optimization problem is determining the optimal stages for revising each decision. While these optimal stages are determined in the first stage, the corresponding revisions are made later at those optimal stages. Furthermore, we prove the NP-hardness of adaptive multistage stochastic programming and establish its connection to classic models such as two-stage stochastic programming and multistage stochastic programming.

- To address the complexity of NACs in adaptive multistage stochastic programming problems, we present theoretical properties that reduce the problem size significantly. In addition, we propose a tailored cutting plane-based solution algorithm that customizes the integer L-shaped method and Benders decomposition to be compatible with mixed-integer master problem and subproblems. Furthermore, we provide a preprocessing approach that eliminates a set of candidate feasible solutions through a cut-generation procedure. Our findings show that these strategies significantly enhance the computational efficiency of decision-making processes.

- We conduct computational experiments on adaptive stochastic programming variants of two classical optimization problems; stochastic lot-sizing and generation expansion planning. Despite the limited flexibility, we show that it is possible to converge to the objective of multistage stochastic programming if revisions are made at optimal stages. It is remarkable that optimal stage selection in less flexible case can be more efficient than arbitrary selection in more flexible case. We conclude that adaptive multistage stochastic programming offers businesses practically feasible solutions that ensure decision commitment in their decision-making process while staying flexible to act upon realized uncertainty at the most critical stages.

- The generation expansion problem is calibrated with real-world data. Through this analysis, we provide managerial insights on optimal revision stages depending on cost and technical parameters of generation sources. We further show the computational efficiency of our reformulations and solution algorithms, where our NACs elimination techniques reduce computation times by 59%, and our solution algorithm further reduces it by another 56% on average considering various instances.

The remainder of the paper is organized as follows. Section 2 introduces our Adaptive Multistage Stochastic Programming approach. Section 3 presents a series of NACs elimination techniques to enhance the proposed model. Section 4 provides the solution approach. Afterward, Section 5 shows the value of adaptive multistage stochastic programming by computational experiments on

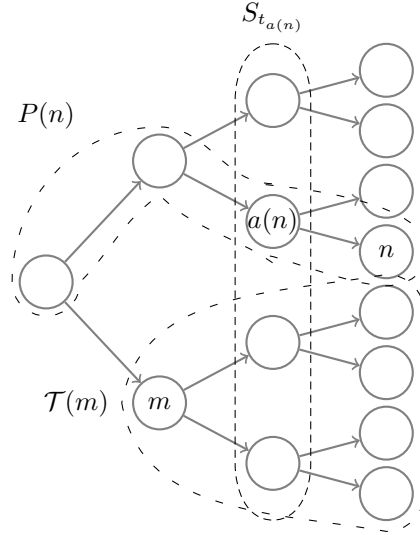
two canonical optimization problems; stochastic lot-sizing and generation expansion planning with managerial insights. Section 6 concludes our paper with final remarks.

2. Adaptive Multistage Stochastic Programming

In this section, we introduce Adaptive Multistage Stochastic Programming (AMSP) for sequential decision-making problems under uncertainty where an optimal balance between decision flexibility and commitment is required. AMSP limits the number of revisions throughout the planning horizon and optimally determines stages at which decision revisions will be allowed for a subset of decisions requiring partial flexibility. Consequently, these decisions can be revised only at those selected stages.

To clarify this new approach, we first present multistage and two-stage stochastic programming formulations, which allow for different flexibility levels for these decisions. In particular, multistage stochastic programming offers a fully flexible setting that allows for revisions as new information is revealed through the planning horizon. In contrast, the two-stage stochastic programming commits a subset of decisions, i.e., here-and-now decisions, statically upfront in planning and does not allow revising these decisions. To this end, AMSP can be interpreted as a more generic approach since it captures both multistage and two-stage stochastic programming depending on the flexibility level allowed in decision-making. Combining the above, we present our AMSP approach in the following order. We first define the underlying stochastic process based on a scenario tree, which is a fundamental approach for representing uncertainty in sequential decision-making problems (Ruszczyński and Shapiro 2003). We then provide two generic formulations for multistage and two-stage stochastic programming and show the associated decision structures on how decisions are revised over the scenario tree. Next, we introduce our AMSP formulation and show its decision structure in comparison with two-stage and multistage stochastic programming. Lastly, we discuss several theoretical properties and prove NP-hardness of AMSP.

We consider a scenario tree \mathcal{T} to model the uncertainty structure throughout planning horizon of T stages. We represent each node of the scenario tree as $n \in \mathcal{T}$, and each stage as $t \in \mathcal{P} := \{1, \dots, T\}$. We define the set of nodes in each stage $t \in \mathcal{P}$ as S_t , and the stage of a node n as $t_n \in \mathcal{P}$. Each node in the scenario tree, except for the leaf nodes, has B branches that represent possible outcomes in the subsequent stage. Each node n , except the root node, has an ancestor denoted by $a(n)$. The unique path from the root node to a specific node n is represented by $P(n)$. Each path from the root node to a leaf node corresponds to a scenario, indicating that every $P(n)$ represents a scenario when $n \in S_T$. We denote the sub-tree rooted at node n until stage t as $\mathcal{T}(n, t)$ for $t_n \leq t \leq T$. To shorten the notation, when the last stage of the sub-tree is T , we let $\mathcal{T}(n) := \mathcal{T}(n, T)$ for all $n \in \mathcal{T}$. The probability of observing each node n is given by p_n , where $\sum_{n \in S_t} p_n = 1$ for all $t \in \mathcal{P}$. Figure 1 shows an example scenario tree to illustrate these concepts.

Figure 1 Scenario Tree with $T = 4$ Stages and $B = 2$ Branches

The decision variables corresponding to node $n \in \mathcal{T}$ consist of the *state* variables $\{\mathbf{x}_n\}_{n \in \mathcal{T}}$, and the *stage* variables $\{\mathbf{y}_n\}_{n \in \mathcal{T}}$. In particular, state variables carry information between different stages, whereas the stage variables at node $n \in \mathcal{T}$ are local variables to their associated stage $t_n \in \mathcal{P}$. We let the dimensions of the variables $\mathbf{x}_n = (x_{1n}, \dots, x_{In})$ and $\mathbf{y}_n = (y_{1n}, \dots, y_{Jn})$ be I and J for all $n \in \mathcal{T}$, respectively, and define the sets $\mathcal{I} = \{1, \dots, I\}$, and $\mathcal{J} = \{1, \dots, J\}$.

Here, we consider a generic formulation where these variables can be integer or continuous. The parameters of each node $n \in \mathcal{T}$ are represented as $(\mathbf{a}_n, \mathbf{b}_n, \mathbf{C}_{nm}, \mathbf{D}_n, \mathbf{d}_n)$. For a given scenario tree, we define multistage stochastic programming (MSP), where both state and stage decisions can be revised in each stage based on the observed uncertainty, as follows:

$$(MSP) \min_{\mathbf{x}, \mathbf{y}} \sum_{n \in \mathcal{T}} p_n (\mathbf{a}_n^\top \mathbf{x}_n + \mathbf{b}_n^\top \mathbf{y}_n) \quad (1a)$$

$$\text{s.t.} \quad \sum_{m \in P(n)} \mathbf{C}_{nm} \mathbf{x}_m + \mathbf{D}_n \mathbf{y}_n \geq \mathbf{d}_n \quad \forall n \in \mathcal{T} \quad (1b)$$

$$\mathbf{x}_n \in \mathcal{X}_n, \mathbf{y}_n \in \mathcal{Y}_n, \quad \forall n \in \mathcal{T} \quad (1c)$$

Here, objective (1a) scales the cost \mathbf{a}_n and \mathbf{b}_n of \mathbf{x}_n and \mathbf{y}_n , respectively, with the probability p_n of observing node $n \in \mathcal{T}$. Constraints (1b) link state and stage variables, and constraints (1c) impose local constraints over each node $n \in \mathcal{T}$ on the variables \mathbf{x}_n and \mathbf{y}_n by the sets \mathcal{X}_n and \mathcal{Y}_n . In formulation (MSP), we allow state variables \mathbf{x}_n to differ in each node $n \in \mathcal{T}$.

An often-used approximation to multistage stochastic programming, called two-stage stochastic programming, results from imposing equality of the state variables \mathbf{x}_n in each stage $t_n \in \mathcal{P}$. This restriction implies that state decisions cannot adapt to realized uncertainty, making it less flexible than formulation (MSP). Note that, in two-stage stochastic programming, uncertain parameters

are typically defined over scenarios which are represented as paths within the scenario tree. However, as in Huang and Ahmed (2009), we define these parameters at the node level instead of the scenario level. The formulation of the two-stage stochastic programming is given as follows:

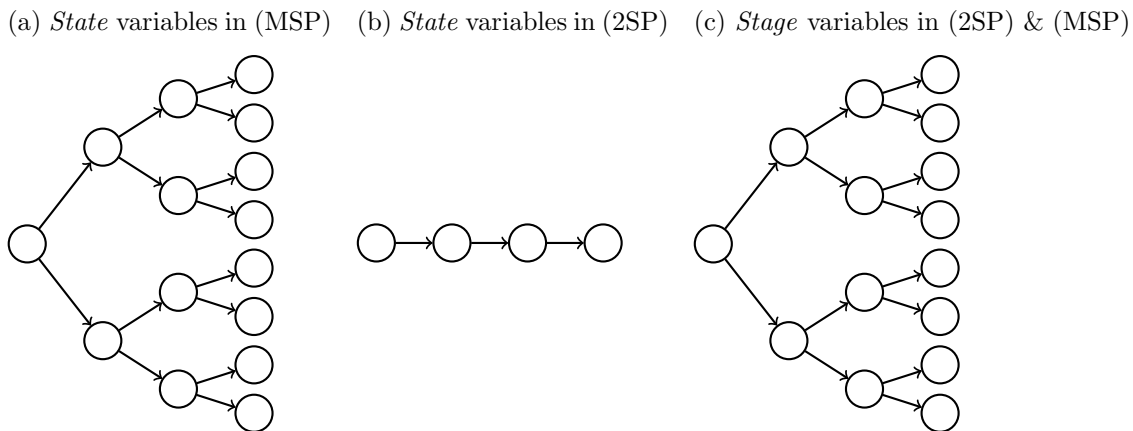
$$(2SP) \min_{\mathbf{x}, \mathbf{y}} \sum_{n \in \mathcal{T}} p_n (\mathbf{a}_n^\top \mathbf{x}_n + \mathbf{b}_n^\top \mathbf{y}_n) \quad (2a)$$

$$\text{s.t.} \quad (1b), (1c)$$

$$\mathbf{x}_m = \mathbf{x}_n \quad \forall m, n \in S_t, t \in \mathcal{P} \quad (2b)$$

In formulation (2SP), objective (2a), and constraints (1b) and (1c) are the same as in formulation (MSP), whereas, with constraints (2b), we impose equality of state variables across different nodes of the same stage, which can be referred as NACs. Before introducing AMSP, we demonstrate the differences in decision structures between formulations (MSP) and (2SP).

Figure 2 Decision structures of state and stage variables



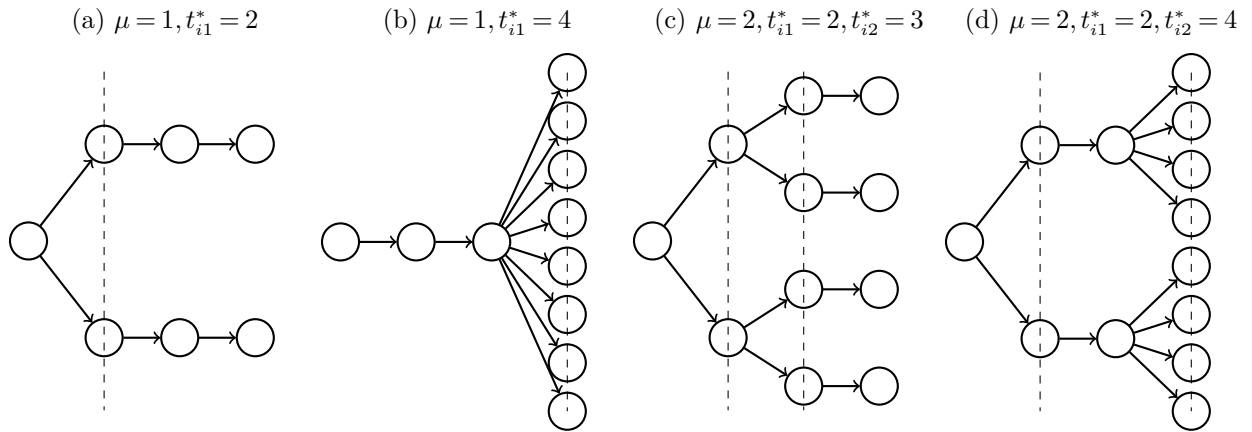
Example 1 (Decision structures in formulations (MSP) and (2SP)) We illustrate decision structures over state variables $\{\mathbf{x}_n\}_{n \in \mathcal{T}}$ and stage variables $\{\mathbf{y}_n\}_{n \in \mathcal{T}}$ over a scenario tree with four stages and two branches (e.g., the scenario tree in Figure 1) in Figure 2. In multistage stochastic programming, in Figure 2a, state variables can be determined for each node $n \in \mathcal{T}$ as uncertainty is realized over the planning horizon. In two-stage stochastic programming, in Figure 2b, the values of the state variables for each stage are determined at the beginning of the planning horizon for $t = 1, \dots, T$, and cannot be adjusted for each realization. However, the decision structure of the stage variables $\{\mathbf{y}_n\}_{n \in \mathcal{T}}$ are the same under both approaches, and they can be determined as uncertainty is revealed in each stage as depicted in Figure 2c.

A decision-maker can desire a balance between decision flexibility and commitment, which requires more flexibility than formulation (2SP) and less flexibility than formulation (MSP) in

revising state variables. To address this need, AMSP limits the change of each state variable over the multi-period planning horizon and includes the decisions on *when* the state variables are allowed to be updated within planning. We refer to the stage at which a state variable can be changed based on uncertainty observation as *revision point*, and the changes themselves as *revisions*. Crucial for AMSP is that the revision points are part of the optimization problem as here-and-now decisions and decided before uncertainties are realized. State variables can only be revised at these revision points, while stage variables can be revised at each stage. Another notable advantage of AMSP is its capability to provide revision points specific to each state variable.

To ensure commitment over state decisions, we limit flexibility by allowing at most μ revision points, where $\mu \in \{0, 1, \dots, T-1\}$. We define a set of revision points for each state variable, which are denoted by $t_{ik}^* \in \{2, \dots, T\}$, representing the k^{th} revision time of variable \mathbf{x}_i for every $i \in \mathcal{I}$ and $k \in \{1, \dots, \mu\}$, which are optimally determined by the adaptive multistage program. We note that $t_{ik}^* < t_{i(k+1)}^*$ should be satisfied for every $k = 1, \dots, \mu - 1$, as $(k+1)^{\text{st}}$ revision can only occur after k^{th} revision takes place. Before introducing the formulation of AMSP, we illustrate various decision structures of the state variables under different revision points.

Figure 3 Decision structures of state variable i for AMSP



Example 2 (Decision structures of state variables in AMSP) We highlight how AMSP works by showing four examples of decision structures under given revision point decisions in Figure 3. We consider a state variable $\{x_{in}\}_{n \in \mathcal{T}}$, for a given i , considering the scenario tree in Figure 1. In Figure 3a and 3b, μ is set to 1, which implies that only one revision is allowed for the state variable $\{x_{in}\}_{n \in \mathcal{T}}$. In Figure 3a, the second stage is determined as the revision stage, and decision variables corresponding to the second stage's nodes can differ based on the realized uncertainty in stage two. However, the nodes for the third and fourth stages at each sub-tree rooting from nodes of the second stage are condensed, meaning that a single decision is made for those condensed nodes.

In other words, one can commit to decisions made for third and fourth stages in the second stage. In Figure 3b, the revision stage is the fourth stage. Thus, nodes at the first three stages are condensed until this time. Furthermore, Figures 3c and 3d present cases where μ is set to 2, where the decisions are revised in the second and third stages and the second and fourth stages, respectively.

We formulate AMSP as a stochastic program considering μ revision points for each state variable over the planning horizon. We include revision points $\Gamma_i^* = \{t_{i1}^*, \dots, t_{i\mu}^*\}, i \in \mathcal{I}$ as decision variables by imposing equalities between the state variables depending on these revision points, leading to the following formulation:

$$\min_{\mathbf{x}, \mathbf{y}, \mathbf{t}^*} \sum_{n \in \mathcal{T}} p_n (\mathbf{a}_n^\top \mathbf{x}_n + \mathbf{b}_n^\top \mathbf{y}_n) \quad (3a)$$

$$\text{s.t.} \quad (1b), (1c)$$

$$x_{im} = x_{in} \quad \forall m, n \in S_t, \quad t < t_{i1}^*, \quad i \in \mathcal{I} \quad (3b)$$

$$x_{im} = x_{in} \quad \forall m, n \in S_t \cap \mathcal{T}(l, t_{i(k+1)}^*), \quad l \in S_{t_{ik}^*}, \quad t_{i(k+1)}^* > t \geq t_{ik}^*, \quad \mu - 1 \geq k \geq 1, \quad i \in \mathcal{I} \quad (3c)$$

$$x_{im} = x_{in} \quad \forall m, n \in S_t \cap \mathcal{T}(l), \quad l \in S_{t_{i\mu}^*}, \quad t \geq t_{i\mu}^*, \quad i \in \mathcal{I} \quad (3d)$$

$$t_{ik}^* \in \Gamma_i^* \subseteq \{2, \dots, T\} \quad \forall i \in \mathcal{I}, \quad k \in \{1, \dots, \mu\} \quad (3e)$$

Objective (3a), constraints (1b) and (1c) are the same as in formulations (2SP) and (MSP). Constraints (3b)–(3d) capture the adaptive multistage relationship under the revision decisions through the NACs tailored for this problem for controlling the decision structures of the nodes of the scenario tree. Specifically, constraints (3b) and (3d) ensure the equality of state variables up to the first and after the last revision point, respectively, depending on the subtrees rooted at the revision points. Similarly, constraint (3c) imposes equality of state variables between corresponding nodes for the remaining revision points.

Formulation (3) is a nonlinear stochastic program since constraints (3b)–(3d) depend on the set Γ_i^* that involves revision points for every $i \in \mathcal{I}$. To linearize the formulation, we introduce an auxiliary integer variable r_{it} for each $i \in \mathcal{I}$ and $t \in \mathcal{P}$, which indicates the total number of revision points of the i^{th} state variable until stage t . For example, in Figure 3a, $r_{i1} = 0$ and $r_{i2} = r_{i3} = r_{i4} = 1$ because the state variables are revised for the first time at stage two, and there are no revisions after that. In Figure 3b, $r_{i1} = r_{i2} = r_{i3} = 0$ and $r_{i4} = 1$ since the first revision point is at stage four. In Figures 3c and 3d, the state variables are revised twice, and the corresponding auxiliary variables are $r_{i1} = 0, r_{i2} = 1, r_{i3} = r_{i4} = 2$, and $r_{i1} = 0, r_{i2} = r_{i3} = 1, r_{i4} = 2$, respectively. By leveraging the auxiliary revision variable r_{it} , we can reformulate the NACs (3b)–(3d) in a linear fashion. Let \bar{x}_i be the upper bound on the state variable for $i \in \mathcal{I}$. If there is no revision between stages t and t' where $t' \geq t \in \mathcal{P}$ (i.e., $r_{it} = r_{it'}$), then the NACs enforce equality of state variables at any node pair

x_{im}, x_{in} at stage t' of the subtree $T(l)$ rooting from the node l of stage t for every $m, n \in S_{t'} \cap \mathcal{T}(l)$. On the other hand, if there is a revision, then these constraints become inactive and serve only as an upper bound on the state variables. The generic linear formulation for AMSP with at most μ revisions is then given as follows.

$$(AMS_\mu) \min_{\mathbf{x}, \mathbf{y}, \mathbf{r}} \sum_{n \in \mathcal{T}} p_n (\mathbf{a}_n^\top \mathbf{x}_n + \mathbf{b}_n^\top \mathbf{y}_n) \quad (4a)$$

$$\text{s.t.} \quad (1b), (1c)$$

$$x_{im} \geq x_{in} - \bar{x}_i (r_{it'} - r_{it}) \quad \forall m, n \in S_{t'} \cap \mathcal{T}(l), l \in S_t, t' \geq t, t \in \mathcal{P}, i \in \mathcal{I} \quad (4b)$$

$$x_{im} \leq x_{in} + \bar{x}_i (r_{it'} - r_{it}) \quad \forall m, n \in S_{t'} \cap \mathcal{T}(l), l \in S_t, t' \geq t, t \in \mathcal{P}, i \in \mathcal{I} \quad (4c)$$

$$r_{i1} \leq r_{i2} \leq \dots \leq r_{iT} \leq \mu \quad \forall i \in \mathcal{I} \quad (4d)$$

$$r_{i(t+1)} - r_{it} \leq 1 \quad \forall t \in \mathcal{P} \setminus \{T\}, i \in \mathcal{I} \quad (4e)$$

$$r_{i1} = 0 \quad \forall i \in \mathcal{I} \quad (4f)$$

$$r_{it} \in \mathbb{Z}_+, \forall t \in \mathcal{P}, i \in \mathcal{I} \quad (4g)$$

In this formulation, objective (4a), constraints (1b) and (1c) are similar to those in formulation (3). Constraints (4b) and (4c) ensure non-anticipativity and shape the decision structure for state variables. Constraint (4d) restricts the total number of revisions and ensures that revision variables are in non-decreasing order. Furthermore, constraints (4e)–(4g) maintain the special structure of the integer revision variables such that there is no revision at the beginning of the planning and there can be at most one revision between consecutive stages.

We demonstrate that more revisions on the state decisions enhances the flexibility of the model, and we provide a formalization of this observation in the subsequent lemma.

Lemma 1 *For (AMS_μ) , there exist an optimal solution where $r_{iT} \leq \mu$ is binding for all $i \in \mathcal{I}$.*

Proof: See Appendix A.1. \square

In the following proposition, we formalize the relationship between formulation (AMS_μ) and existing optimization approaches in stochastic programming literature.

Proposition 1 *For $\mu = 0$, formulation (AMS_μ) is equivalent to two-stage stochastic programming formulation (2SP). For $\mu = T - 1$, formulation (AMS_μ) is equivalent to multistage stochastic programming formulation (MSP).*

Proof: See Appendix A.2. \square

Proposition 1 highlights that formulation (AMS_μ) is a generalization of existing stochastic programming approaches. Furthermore, when μ is set to 1, formulation reduces to the adaptive two-stage stochastic programming proposed in Basciftci et al. (2019).

By increasing μ , we can enhance the flexibility in decision-making, leading to improvements in the objective value. We formalize this in the following proposition, where we let $z(\cdot)$ denote the optimal objective value of a given problem. For example, $z(AMS_\mu)$ is the optimal objective value of formulation (AMS_μ) . It is important to note that this study assumes that formulation $(2SP)$ is feasible and consequently (AMS_μ) is feasible under any μ value.

Proposition 2 $z(2SP) = z(AMS_0) \geq z(AMS_1) \geq \dots \geq z(AMS_{(T-2)}) \geq z(AMS_{(T-1)}) = z(MSP)$.

Proof: See Appendix A.3. \square

We next analyze the complexity of this problem and formalize it in the following proposition.

Proposition 3 *AMSP is NP-Hard.*

Proof: See Appendix A.4. \square

A special case of AMSP arises when the state variables are binary. This characteristic permits the relaxation of the integrality requirement for the revision decision variables \mathbf{r} . We formalize this observation in Proposition 4.

Proposition 4 *Let the state variables be binary with $x_{in} \in \{0, 1\}$ for all $i \in \mathcal{I}$, $n \in \mathcal{T}$. Then, the integrality conditions on the revision decision variables r_{it} for all $i \in \mathcal{I}$ and $t \in \mathcal{P}$ can be relaxed within (AMS_μ) .*

Proof: See Appendix A.5. \square

3. Elimination Techniques for Non-anticipativity Constraints

The size of formulation (AMS_μ) increases exponentially with the increasing number of stages, making the model challenging to solve. In this section, we enhance formulation (AMS_μ) by identifying a series of constraint elimination techniques for the NACs (4b) and (4c). Recall that these constraints consider any combination of stages $t' \geq t \in \mathcal{P}$, and impose equality between nodes m, n at stage t' if there exists a subtree rooted at time t containing both nodes m and n when there is no revision between t and t' . In the following, we first present a series of propositions that identify how we can reduce the number of NACs, and then combine these results to effectively reformulate (AMS_μ) .

The first elimination, presented in Proposition 5, results from imposing a cyclic relation between the nodes associated with constraints (4b) and (4c). Specifically, instead of considering each m, n

pair in the subtree sets $S_{t'} \cap \mathcal{T}(l)$ for every $l \in S_t$, we order the nodes in these sets for each t' , and enforce equality between these ordered pair of nodes.

Proposition 5 *Let $t \in \mathcal{P}$, $t' \geq t$, and $l \in S_t$ be given. Let $K = |S_{t'} \cap \mathcal{T}(l)|$. Without loss of generality, we can order the nodes in $S_{t'} \cap \mathcal{T}(l)$ as $(n, n+1, \dots, n+K-1)$. Then, it is sufficient to include constraints (4b) and (4c) only for the consecutive node pairs (e.g., $(n, n+1), (n+1, n+2), \dots, (n+K-1, n)$) from the set $S_{t'} \cap \mathcal{T}(l)$ as follows:*

$$x_{in} \geq x_{i(n+1)} - \bar{x}_i (r_{it'} - r_{it}) \quad n \in S_{t'} \cap \mathcal{T}(l), \quad l \in S_t, \quad t' \geq t, \quad t \in \mathcal{P}, \quad i \in \mathcal{I} \quad (5)$$

We note that we should remove the ordered node $n+K$ from the set $S_{t'} \cap \mathcal{T}(l)$ and add the constraint between $x_{i(n+K-1)}$ and x_{in} , but to enhance readability we slightly abuse notation in constraints (5) by using node $n+K$ for node n .

Proof: See Appendix A.6. \square

In Proposition 5, we focus on node pairs (m, n) that belong to the same stage t' , and remove redundant constraints by exploiting an arbitrary ordering argument between such nodes. Alternatively, we observe that any given node pair (m, n) belonging to the set $S_{t'}$ may be included in various NACs of different (t, t') combinations. However, it is sufficient to link a node pair (m, n) depending on only one particular (t, t') combination, where t is the stage of the last common ancestor of nodes (m, n) . This observation is based on the non-decreasing nature of the r variables as formalized below.

Proposition 6 *Let $t_{a(m,n)}$ be the stage of the last common ancestor of node pair (m, n) from the set S_t of stage t . It is sufficient to include NACs of (m, n) only for $t, t_{a(m,n)}$ combination, and accordingly, constraints (4b) and (4c) can be replaced by the following constraints:*

$$x_{im} \geq x_{in} - \bar{x}_i \left(r_{it} - r_{it_{a(m,n)}} \right) \quad \forall m, n \in S_t, t \in \mathcal{P}, i \in \mathcal{I} \quad (6a)$$

$$x_{im} \leq x_{in} + \bar{x}_i \left(r_{it} - r_{it_{a(m,n)}} \right) \quad \forall m, n \in S_t, t \in \mathcal{P}, i \in \mathcal{I} \quad (6b)$$

Proof: See Appendix A.7. \square

In Proposition 6, we include only one pair of NACs for any (m, n) node pair in the same stage. Furthermore, we observe that NACs are not required for certain (m, n) pairs, depending on the value of μ . This observation is based on the fact that the model revises state decisions to increase flexibility as much as possible, resulting in $r_{iT}^* = \mu$ for $i \in \mathcal{I}$ at an optimal solution, as shown in Lemma 1.

Proposition 7 Let $t \in \{1, \dots, T\}$ and $t' \geq t$. For given μ , NACs between a particular (t', t) combination become redundant, if $t' > t + (T - \mu) - 1$. Then, constraints (4b) and (4c) reduce to:

$$x_{im} \geq x_{in} - \bar{x}_i (r_{it'} - r_{it}) \quad \forall m, n \in S_{t'} \cap \mathcal{T}(l), l \in S_t, \min(t + (T - \mu) - 1, T) \geq t' \geq t, t \in \mathcal{P}, i \in \mathcal{I} \quad (7a)$$

$$x_{im} \leq x_{in} + \bar{x}_i (r_{it'} - r_{it}) \quad \forall m, n \in S_{t'} \cap \mathcal{T}(l), l \in S_t, \min(t + (T - \mu) - 1, T) \geq t' \geq t, t \in \mathcal{P}, i \in \mathcal{I} \quad (7b)$$

Proof: See Appendix A.8. \square

We remark that Propositions 5-7 do not dominate each other and thus the number of NACs reduces further if we combine the propositions as follow:

$$x_{in} \geq x_{i(n+1)} - \bar{x}_i \left(r_{it} - r_{it_{a(n,n+1)}} \right) \quad \forall n \in S_t, t_{a(n,n+1)} \in \mathcal{P}_t, t \in \mathcal{P}, i \in \mathcal{I} \quad (8)$$

where set \mathcal{P}_t is defined as $\{(t - (T - \mu) + 1)^+, \dots, t - 1\}$ for a given t . We summarize the main outcome of this section in Theorem 1.

Theorem 1 *AMSP can be correctly formulated as follows:*

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{y}, \mathbf{r}} \quad & \sum_{n \in \mathcal{T}} p_n (\mathbf{a}_n^\top \mathbf{x}_n + \mathbf{b}_n^\top \mathbf{y}_n) \\ \text{s.t.} \quad & (1b), (1c), (4d) - (4g), (8) \end{aligned} \quad (9a)$$

We refer to this formulation as (AMS_μ^+) , which is equivalent to formulation (AMS_μ) .

We demonstrate the impact of Propositions 5–7 on the number of necessary NACs by considering an example scenario tree with ten stages ($T = 10$) and two branches ($B = 2$) focusing on a single state variable ($I = 1$). Table 1 displays the required number of NACs for each (t, t') pair in formulation (AMS_μ) which does not include the proposed elimination techniques. We observed a total of 688810 NACs, which has notable implications for the complexity of the model.

Table 2 displays the resulting changes in the number of included NACs when we apply Propositions 5-7. Applying Proposition 5 alone reduces the total number of NACs to 8194 by imposing a cyclic relationship on the nodes rather than having NACs for every possible combination of m, n pairs in stage t' . Consequently, the number of NACs for every t' equals the number of nodes in that stage. By applying both Propositions 5 and 6, we further reduce the total number of NACs to 2026. Prior to this, a node pair could be seen in many constraints, but now it is unique to a particular t, t' pair. Depending on the maximum allowed number of revisions μ , we can further decrease the number of constraints by considering Proposition 7. We display an example where μ is set to 4. In this case, 52 more constraints can be eliminated. The number of eliminated constraints changes to 2, 8, 22, 114, 240, 494, and 1004 when μ is set to be 1, 2, 3, 5, 6, 7, and 8, respectively.

Table 1 Number of NACs for formulation (AMS_μ) with $B = 2$ Branches

$t \backslash t'$	2	3	4	5	6	7	8	9	10
1	2	12	56	240	992	4032	1.6e4	6.5e4	2.6e5
2		4	24	112	480	1984	8064	3.2e4	1.3e5
3			8	48	224	960	3968	1.6e4	6.5e4
4				16	96	448	1920	7936	3.2e4
5					32	192	896	3840	1.6e4
6						64	384	1792	7680
7							128	768	3584
8								256	1536
9									512

Table 2 Number of NACs with Propositions 5 - 7

$t \backslash t'$	Proposition 5									Propositions 5 and 6									Propositions 5, 6, and 7 ($(AMS_{\mu+})$)								
	2	3	4	5	6	7	8	9	10	2	3	4	5	6	7	8	9	10	2	3	4	5	6	7	8	9	10
1	2	4	8	16	32	64	128	256	512	2	2	2	2	2	2	2	2	2	2	2	2	2	2				
2		4	8	16	32	64	128	256	512		4	4	4	4	4	4	4	4		4	4	4	4	4			
3			8	16	32	64	128	256	512			8	8	8	8	8	8	8			8	8	8	8	8		
4				16	32	64	128	256	512				16	16	16	16	16	16				16	16	16	16	16	
5					32	64	128	256	512					32	32	32	32	32					32	32	32	32	32
6						64	128	256	512						64	64	64	64						64	64	64	64
7							128	256	512							128	128	128							128	128	128
8								256	512								256	256								256	256
9									512									512									512

The total number of NACs for formulation (AMS_μ) can be calculated as $I \times \sum_{t=1}^{T-1} \sum_{j=1}^{T-t} B^{t-1} \binom{B^j}{2} \times 2$. However, after applying all of the propositions, the total number of NACs in formulation $(AMS_{\mu+})$ reduces $I \times \sum_{t=1}^{T-1} \sum_{j=1}^{\min(T-t, t+T-\mu-1)} B^t$, which represents a substantial decrease in the number of NACs. Table 3 illustrates the total number of NACs for varying numbers of stages and branches when there is a single state variable. Note that Proposition 7 depends on the value of μ , and we present two examples when μ is set to 2 and 4. The propositions lead to a significant reduction in the total number of NACs, ranging from 90% to almost 100%. Note that the total number of NACs in Table 3 for the case of 2 branches, 10 stages, and $\mu = 4$ corresponds to the sum of the NACs listed in Table 2 for $(AMS_{\mu+})$ when all propositions are applied.

Table 3 Total number of NACs over different instance sizes

Branch	2						3					
	5	6	7	8	9	10	5	6	7	8	9	10
Formulation (AMS_μ)	522	2346	1.0e4	4.2e4	1.7e5	6.9e5	1.0e4	9.7e4	8.9e5	8.0e6	7.3e7	6.5e8
Formulation $(AMS_{\mu+})$, $\mu = 2$	44	106	232	486	994	2018	159	522	1614	4893	1.5e4	4.4e4
Formulation $(AMS_{\mu+})$, $\mu = 4$	0	62	188	442	952	1974	0	363	1455	4734	1.4e4	4.4e4

4. Solution Approach

This section introduces a general solution approach for solving AMSPs with mixed-integer state and stage variables. Standard solution approaches for multistage stochastic programming cannot be used due to the presence of integer revision decisions r_{it} for every $i \in \mathcal{I}$ and $t \in \mathcal{P}$. To address this issue, we propose a solution approach that separates these complicating decision variables from the remainder of the problem, which leverages Benders decomposition (Benders 1962) and the integer L-shaped method (Van Slyke and Wets 1969). More specifically, we decompose the problem into a master problem, in which we make the revision decisions, and a subproblem, in which we solve the adaptive multistage problem under the given revision decisions. Then, we present an iterative process, where we first solve the master problem and then add cuts to this problem using the subproblem. Note that integer L-shaped cuts are designed for pure binary master programs, while Benders cuts require convexity of the subproblem. In order to accommodate the presence of integer revision decisions in the master problem and possible mixed-integer subproblems, we make the necessary modifications to incorporate these cuts, as explained later in this section.

First, we present the master problem (10) by introducing an auxiliary variable θ to approximate the cost of the subproblem as follows:

$$(MP) \min_{\mathbf{r}, \theta} \theta \tag{10a}$$

$$\text{s.t. } r_{i1} \leq r_{i2} \leq \dots \leq r_{iT} \leq \mu \quad \forall i \in \mathcal{I} \tag{10b}$$

$$r_{i(t+1)} - r_{it} \leq 1 \quad \forall t \in \mathcal{P} \setminus \{T\}, i \in \mathcal{I} \tag{10c}$$

$$r_{i1} = 0 \quad \forall i \in \mathcal{I} \tag{10d}$$

$$r_{it} \in \mathbb{Z}_+ \quad \forall t \in \mathcal{P}, i \in \mathcal{I} \tag{10e}$$

$$\theta \geq L \tag{10f}$$

$$(r, \theta) \in \Phi \tag{10g}$$

$$(r, \theta) \in \Psi \tag{10h}$$

Here, constraints (10b)-(10e) correspond to the integer revision point related constraints (4d)-(4g) as in models (4) and (9). Constraint (10f) sets an initial lower bound on the variable θ . To set this lower bound L , we solve the multistage problem as a preprocessing step since the objective of the multistage stochastic program provides a lower bound on the adaptive multistage problem. Sets Φ and Ψ in constraints (10g) and (10h) correspond to the sets of L-shaped and Benders optimality cuts, respectively. We incorporate only optimality cuts and omit feasibility cuts, given our assumption that the AMSP is feasible under any revision vector \mathbf{r} (see Proposition 2).

We let $\bar{\mathbf{r}}$ denote the vector of revision decisions obtained from the master problem. Then, using the improved formulation of the adaptive multistage problem ($AMS_{\mu,+}$), the subproblem can be defined as follows:

$$(SP) \quad \mathcal{Q}(\bar{\mathbf{r}}) = \min_{\mathbf{x}, \mathbf{y}} \sum_{n \in \mathcal{T}} p_n (\mathbf{a}_n^\top \mathbf{x}_n + \mathbf{b}_n^\top \mathbf{y}_n) \quad (11a)$$

$$\text{s.t.} \quad \sum_{m \in P(n)} \mathbf{C}_{nm} \mathbf{x}_m + \mathbf{D}_n \mathbf{y}_n \geq \mathbf{d}_n \quad \forall n \in \mathcal{T} \quad (11b)$$

$$\mathbf{x}_n \in \mathcal{X}_n, \mathbf{y}_n \in \mathcal{Y}_n, \quad \forall n \in \mathcal{T} \quad (11c)$$

$$x_{in} \geq x_{i(n+1)} - \bar{x}_i \left(\bar{r}_{it} - \bar{r}_{it_{a(n,n+1)}} \right) \quad \forall n \in \mathcal{S}_t, t_{a(n,n+1)} \in \mathcal{P}_t, t \in \mathcal{P}, i \in \mathcal{I} \quad (11d)$$

The integer L-shaped method can be applied to problems where the master problem includes pure binary decision variables (Laporte and Louveaux 1993). However, our master problem includes integer revision decisions. One potential solution is representing these integer variables as a sum of binary decision variables. This representation allows expressing the optimality cuts obtained from the subproblem in terms of these auxiliary binary variables, which introduces additional complexity to the solution procedure. Instead, we propose an alternative scheme by utilizing the special structure of the revision decisions. Specifically, the difference in revision points between two consecutive stages, denoted as $(r_{it} - r_{i(t-1)})$ for $t \in \mathcal{P} \setminus \{1\}$ and $i \in \mathcal{I}$, can only be one or zero. If this value is one, then there is a revision at time t for the state variable i , and zero, otherwise. We utilize this information to define the set $Y_i(\bar{\mathbf{r}}) := \{t \in \mathcal{P} : \bar{r}_{it} - \bar{r}_{i(t-1)} = 1\}$ under a given revision vector $\bar{\mathbf{r}}$, representing the set of periods with revision for the state variable i . Accordingly, using the subproblem (11), the integer L-shaped based optimality cuts can be obtained as follows:

$$\theta \geq (\mathcal{Q}(\bar{\mathbf{r}}) - L) \left(\sum_{i \in \mathcal{I}} \sum_{t \in Y_i(\bar{\mathbf{r}})} (r_{it} - r_{i(t-1)} - 1) - \sum_{i \in \mathcal{I}} \sum_{t \notin Y_i(\bar{\mathbf{r}})} (r_{it} - r_{i(t-1)}) \right) + \mathcal{Q}(\bar{\mathbf{r}}) \quad (12)$$

Since set Φ contains exponentially many cuts in the form of (12), obtaining these cuts requires solving mixed-integer subproblems, which may cause a computational burden. Therefore, we integrate Benders optimality cuts to decrease the feasible region of the master problem. As convexity of the subproblems is necessary for Benders decomposition to converge to an optimal solution and derive optimality cuts (Geoffrion 1972), we convexify the subproblem by relaxing the integrality of the state and stage variables. We refer to the relaxed version of formulation (SP) as (RSP) and denote its objective function as $\underline{Q}(\bar{\mathbf{r}})$. Accordingly, by taking the dual of this relaxed subproblem, we obtain Benders optimality cuts Ψ as follows:

$$\theta \geq \sum_{n \in \mathcal{T}} \mathbf{d}_n^\top \bar{\mathbf{\Lambda}}_n - \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{P}} \sum_{n \in \mathcal{S}_t} \sum_{t_{a(n,n+1)} \in \mathcal{P}_t} \bar{x}_i \left(r_{it} - r_{it_{a(n,n+1)}} \right) \bar{\Pi}_{itn(n+1)} \quad (13)$$

where $\bar{\mathbf{\Lambda}}$ and $\bar{\mathbf{\Pi}}$ represents the optimal values of the dual variables corresponding to constraints (11b) and (11d), respectively, under the solution $\bar{\mathbf{r}}$. For simplicity in notation, we assume that constraints (11c) are non-negativity restrictions, and we omit their corresponding dual variables in illustrating (13). Solving the dual of (RSP) and obtaining Benders cuts is computationally

easier than solving (SP) and obtaining L-shaped cuts, as it involves linear programs rather than their mixed-integer counterparts. Therefore, we solve (RSP) and incorporate Benders cuts in each iteration as a revision vector is obtained by the master problem. However, the decision to solve (SP) and include L-shaped cuts depends on the improvement observed in the solution of (RSP) using the obtained revision vector. In this way, we selectively add L-shaped cuts in a subset of the iterations for the sake of solution time. The detailed procedure is further explained later in Algorithm 1.

To accelerate the solution algorithm further, we examine the timing of the revision point decisions over the problem instances with different lengths of planning horizons.

Observation 1 *In our preliminary experiments, we consider a modified version of the original problem (AMS_μ) , by reducing the planning horizon by n stages, resulting in a shortened scenario tree denoted as $\mathcal{T}(n_0, T - n)$, where n_0 is the root node of \mathcal{T} . Specifically, we remove the stages and decision variables from $T - n + 1$ to T from the original problem, and solve the reduced version, which we call (AMS_μ^{T-n}) . In comparison to the solution obtained from the original problem (AMS_μ) , we have observed that as the planning horizon is reduced, revision points of the original problem that are scheduled between stages $1, \dots, T - n$ do not shift to subsequent stages in the solution obtained from (AMS_μ^{T-n}) ; instead, they either remain stable or shift to the previous stages.*

We leverage this observation in the preprocessing step of our algorithm to eliminate a set of candidate feasible solutions by solving a simpler variant of the original problem. Therefore, we first solve AMS_μ^{T-n} with a reduced planning horizon for a given $n \in \{1, \dots, T - 1\}$. We then obtain an optimal revision solution from AMS_μ^{T-n} , denoted as $\underline{\mathbf{r}}$. Due to Observation 1, we expect the revision points of the original problem at stages $1, \dots, T - n$ to either remain at the same stage or move to later stages compared to the shorter-term version of this problem. Thus, we propose the heuristic cuts (14) to be added to the master problem.

$$r_{it} \geq \underline{r}_{it} \quad \forall t \in \{1, \dots, T - n\}, i \in \mathcal{I} \quad (14)$$

Combining above, we summarize our solution approach in Algorithm 1. In the first step of this algorithm, we solve the multistage stochastic problem to use its objective as a lower bound for (MP) within constraint (10f). Next, we solve the $(AMS_\mu+)$ problem for $T - n$ stages to obtain heuristic cuts and an initial solution $\underline{\mathbf{r}}$. We then include heuristic cuts to the master problem. In the first step of the while loop, we solve the relaxed subproblem under fixed $\underline{\mathbf{r}}$ to obtain its objective value $z(RSP)$ and generate Benders optimality cuts to be added to Ψ . If the value of $z(RSP)$ is less than or equal to relaxed upper bound (RUB), the upper bound (UB) is updated

Algorithm 1 Solution Algorithm

-
- 1: Set $LB = -\infty$, $UB = RUB = \infty$, $\Psi = \emptyset$, $\Phi = \emptyset$.
 - 2: $L \leftarrow$ Solve (MSP) to obtain $z(MSP)$.
 - 3: Solve (AMS_{μ}^{T-n+}) to obtain $\underline{\mathbf{r}}$.
 - 4: Add heuristic cuts (14) to (MP) .
 - 5: Set $\bar{\mathbf{r}}$ to $\underline{\mathbf{r}}$.
 - 6: **while** $(1 - LB/UB) < \epsilon$ **do**
 - 7: $z(RSP) \leftarrow$ Solve (RSP) subject to $\bar{\mathbf{r}}$.
 - 8: Generate Benders optimality cuts (13) and add them to Ψ .
 - 9: **if** $z(RSP) \leq RUB$ **then**
 - 10: $RUB \leftarrow z(RSP)$.
 - 11: $z(SP) \leftarrow$ Solve (SP) subject to $\bar{\mathbf{r}}$.
 - 12: Generate L-shaped optimality cuts (12) and add them to Φ .
 - 13: **if** $z(SP) \leq UB$ **then**
 - 14: $UB \leftarrow z(SP)$.
 - 15: $\mathbf{r}^* \leftarrow \bar{\mathbf{r}}$.
 - 16: Solve (MP) with Ψ and Φ to obtain $\bar{\mathbf{r}}$.
 - 17: Set $LB \leftarrow z(MP)$.
-

to the value of $z(RSP)$. The algorithm then solves the subproblem under fixed $\underline{\mathbf{r}}$ and generates L-shaped optimality cuts to be added to Φ . If the value of $z(SP)$ is less than or equal to UB , the upper bound is updated to the value of $z(SP)$, and \mathbf{r}^* is set to $\underline{\mathbf{r}}$. We then solve (MP) with the obtained cuts Ψ and Φ to get revisions $\underline{\mathbf{r}}$ and set the lower bound (LB) to $z(MP)$. We repeat this procedure until the gap $(1-LB/UB)$ is smaller than ϵ .

Remark 1 *If the heuristic cut in Step 4 and the condition in Step 9 of Algorithm 1 are excluded, the proposed approach is guaranteed to converge to an optimal solution in a finite number of steps. However, our computational experiments have demonstrated that incorporating these two steps significantly accelerates the algorithm's convergence and results in an optimal solution for almost all instances.*

5. Computational Experiments

We demonstrate the value of AMSP by addressing two distinct multistage stochastic optimization problems that require a balance between decision flexibility and decision commitment. First, in Section 5.1, we study a canonical optimization problem from supply chain management; the stochastic uncapacitated lot-sizing problem. Second, in Section 5.2, we provide a case study of the generation expansion planning problem, which is fundamental in our society's quest toward a

transition to renewable energy under various uncertainties. Besides showing the value of AMSP, we demonstrate the computational performance of the NACs elimination techniques proposed in Section 3 and our solution algorithm presented in Section 4.

All the computational experiments are carried out on an Intel Xeon E5 2680v3 CPU with a 2.5 GHz processor and 32 GB RAM. The implementation is performed in Python, utilizing the Gurobi 9.1.0 solver to solve the mathematical models (4) and (9), as well as the master problem (10) and subproblem (11) within our solution algorithm. For consistency, a time limit of 5 hours is imposed on all experiments.

To evaluate the benefit of AMSP, we introduce a metric called the “Value of AMSP” (VAMS), which compares its solution quality with two-stage (2SP) and multistage stochastic programming (MSP) solutions. Note that two-stage stochastic programming (2SP) offers a feasible solution for AMSP and provides an upper bound, while multistage stochastic programming (MSP) is a relaxation for AMSP, providing a lower bound. Accordingly, we define the VAMS for a given level of flexibility μ as follows:

$$\text{VAMS} = \frac{z(2SP) - z(AMS_\mu)}{z(2SP) - z(MSP)} \times 100\% \quad (15)$$

The VAMS metric quantifies the proximity between the objective values of AMSP and multistage stochastic programming relative to two-stage stochastic programming. Notably, the VAMS value is 0% if $\mu = 0$, as $z(AMS_0)$ is equal to the optimal value $z(2SP)$ of two-stage stochastic programming. Conversely, the VAMS value is 100% if $\mu = T - 1$, as $z(AMS_{T-1})$ is equal to the optimal value $z(MSP)$ of multistage stochastic programming. A higher VAMS implies that AMSP can generate solutions comparable to those of multistage stochastic programming by optimizing revision points.

5.1. Adaptive Multistage Stochastic Lot-Sizing Problem

The stochastic lot-sizing problem (Guan and Miller 2008) concerns determining optimal production schedules of a given number of production sources to cover the demand of a single item over a finite planning horizon. The goal is to minimize the expected sum of production setup, production amount, and inventory holding cost considering stochasticity of demand and cost parameters. The production setup decisions are made first to determine whether to initiate production for each source at a given stage. Based on the setup decisions, the production amount for each source and the associated inventory levels are determined. Multistage stochastic programming is commonly used for lot-sizing problems; it provides flexibility to update all decisions based on the realized uncertainty through the planning horizon. However, in many practical settings, frequent updates to decisions can be detrimental to production scheduling performance, leading to decreased productivity and increased global costs (Herrera et al. 2016). Furthermore, such frequent updates might be even impractical due to various factors, such as dependencies on predetermined schedules for

other items (Kouvelis et al. 2006). Addressing such instabilities while enabling flexible production management is a focus of various studies (Herrera et al. 2016, Meistering and Stadtler 2017). Thus, optimizing the stages at which the decisions will be revised has a significant potential to enhance this problem setting. Therefore, in this section, we introduce the Adaptive Multistage Stochastic Uncapacitated Lot-Sizing Problem.

We consider a set of sources represented by $i \in \mathcal{I}$ and a scenario tree \mathcal{T} that describes the underlying stochastic process. Each scenario in the tree has an equal probability of occurring. The system's uncertainty lies in setup costs α_{in} , production costs β_{in} , inventory holding costs h_n , and demand d_n for every node $n \in \mathcal{T}$. We model the setup decisions as binary variables x_{in} , which take a value of one if there is a setup for production of source i at node n , and zero otherwise. We represent the production amount of source i at node n and the inventory level at node n as continuous variable y_{in} and s_n , respectively. We denote M as a large number for constraint linearization purposes. Note that the setup variables are the *state* decisions, while production and inventory holding variables are the *stage* decisions. Based on the multistage stochastic, single-item, uncapacitated lot-sizing problem formulation from (Guan and Miller 2008), we propose the following AMSP variant:

$$\min \sum_{n \in \mathcal{T}} p_n \left(\sum_{i \in \mathcal{I}} (\alpha_{in} x_{in} + \beta_{in} y_{in}) + h_n s_n \right) \quad (16a)$$

$$\text{s.t. } s_{a(n)} + \sum_{i \in \mathcal{I}} y_{in} = d_n + s_n \quad \forall n \in \mathcal{T} \quad (16b)$$

$$y_{in} \leq M x_{in} \quad \forall n \in \mathcal{T}, i \in \mathcal{I} \quad (16c)$$

$$x_{in} \in \{0, 1\}, y_{in}, s_n \in \mathbb{R}_+ \quad \forall n \in \mathcal{T}, i \in \mathcal{I} \quad (16d)$$

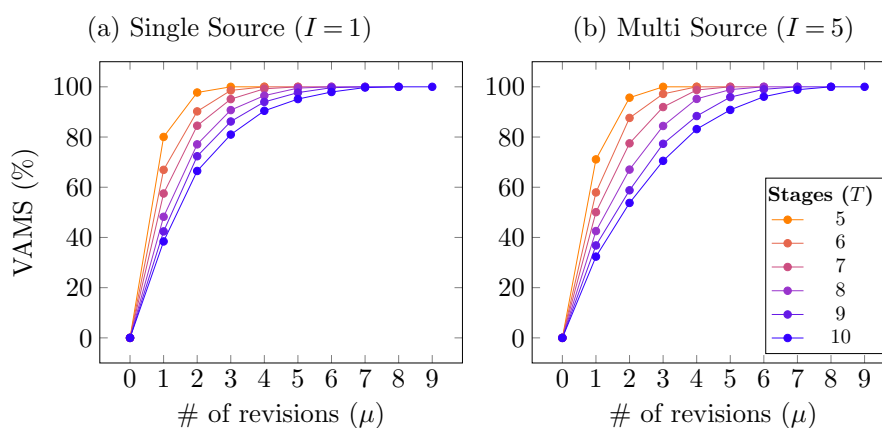
$$(4d) - (4g), (8)$$

Here, objective (16a) minimizes the total expected cost of production, setup, and inventory holding. Constraint (16b) ensures the balance of the inventory flow. Constraint (16c) controls the setup-production relationship. Constraint (16d) defines feasibility sets of the variables. Finally, constraints (4d) - (4g), (8) are the necessary NACs, which also include the revision points \mathbf{r} .

To analyze the stochastic lot-sizing problem described above, we create a scenario tree by generating random cost and demand parameters at each node. Specifically, the setup cost α_{in} is drawn from $\mathcal{U}(100, 250)$, production cost β_{in} is drawn from $\mathcal{U}(0, 5)$, inventory holding cost h_n is drawn from $\mathcal{U}(0, 5)$, and demand d_n is drawn from $\mathcal{U}(0, 100 \times I)$. We create ten different scenario trees and report the results on average.

5.1.1. Potential of AMSP. We examine the potential of AMSP in achieving results comparable to those achieved under full flexibility, despite the presence of decision commitment. Figure 4 shows the effect of increasing the number of revisions μ on the VAMS. Each point represents the VAMS for different values of μ , while the lines represent varying planning horizon lengths T . In Figures 4a and 4b, we explore two cases: one with a single production source ($I = 1$), and the other with multiple production sources ($I = 5$). In each case, the planning horizon length T varies from 5 to 10, and the number of revisions μ is within the range from 0 to $T - 1$.

Figure 4 The performance of AMSP

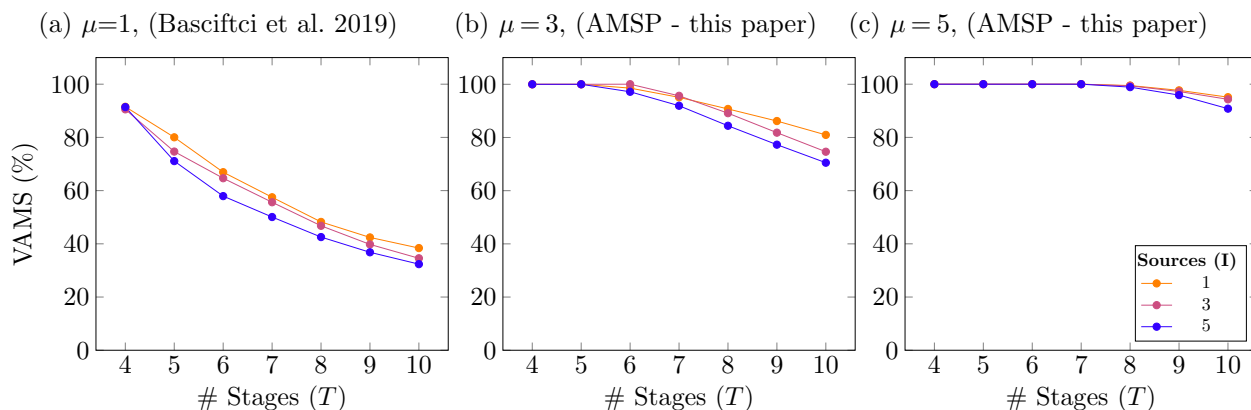


Insight 1 For a fixed planning horizon T , the cost-benefit of increasing the degree of allowed flexibility (i.e., μ) on the Value of the AMSP is diminishing. This implies that in situations where decision commitment is needed in practice, we only need a small number of revision points to obtain most of the value of the fully flexible classic multistage stochastic programming approach.

In Figures 4a and 4b, we observe that as the number of revisions increases, the VAMS improves for all values of T . However, this improvement diminishes as the number of revisions continues to increase. This finding is promising, as it demonstrates that AMSP, even with limited revisions, can achieve comparable results to a fully flexible multistage stochastic programming case. For instance, we obtain a VAMS exceeding 90%, if we revise the decisions twice ($\mu = 2$) in case of a six-stage planning horizon ($T = 6$). This also holds if we revise decisions three times in an eight-stage planning horizon or four times in a ten-stage planning horizon. Furthermore, in specific cases, the VAMS reaches %100 without the need to full flexibility.

5.1.2. Importance of Multiple Revisions. We analyze the potential of considering multiple revision points ($\mu > 1$) during the planning horizon, compared to considering a single revision point ($\mu = 1$) as introduced by Basciftci et al. (2019). Figure 5 show how the VAMS changes as the length of the planning horizon increases under different numbers of sources in the lot-sizing problem. In Figures 5a, 5b, and 5c, μ is fixed to 1, 3, and 5, respectively. Each figure includes results for planning horizons with T stages ranging from 4 to 10 and I sources varying among 1, 3, and 5.

Figure 5 Single versus multiple revision points



Insight 2 *AMSP offers a clear benefit compared to adaptive two-stage stochastic programming. Whereas adaptive two-stage stochastic programming falls short in the case of relatively long planning horizons, AMSP remains of high quality.*

Zooming in on Figure 5a, we observe a sharp decline in the VAMS as the planning horizon lengths increase. For instance, the spread of the VAMS, across a varying number of sources, ranges between %90 to %92 in a four-stage planning horizon, whereas this range is between %32 and %38 for a ten-stage planning horizon. Moreover, the VAMS is likely to converge to 0 for even longer planning horizons, turning into a static approach asymptotically. Comparing this against AMSP in case $\mu = 3$, as shown in Figure 5b, we see that the VAMS remains almost 100% until the planning horizons with six stages. More notable, the spread of VAMS is between %70 to %82 for the 10-period planning horizon, significantly higher than the single revision case in Figure 5a. Further increasing the number of revision points makes the VAMS notably higher for longer planning horizons, as seen in Figure 5c.

Our findings show that VAMS is a reliable approach toward long-term decision-making and highlight the crucial importance of allowing multiple and different revision points for each decision variable in achieving high-quality decisions. In other words, revising decisions only once - as

recommended by adaptive two-stage stochastic programming (Basciftci et al. 2019) - falls short as the number of stages increases, unless the decision-maker has very limited flexibility by allowing at most one revision during planning.

5.1.3. Structure of Optimal Revision Points. Optimally choosing revision points is critical in enhancing the efficiency of decision-making. Our observations reveal that the relationship between different revision decisions is not straightforward and requires further analysis. To provide a concrete example, we study an adaptive multistage stochastic uncapacitated lot-sizing problem with a single source and five stages. We enumerate all possible combinations of revision points for different values of μ and solve Model (16) under these revision points. We note that enumeration for instances with multiple sources and longer planning horizons is computationally intractable.

Table 4 VAMS for single source lot-sizing problem under fixed revisions

$\mu = 1$		$\mu = 2$		$\mu = 3$	
Revision points	VAMS (%)	Revision points	VAMS (%)	Revision points	VAMS (%)
2	7	2, 3	78	2, 3, 4	79
3	78	2, 4	79	2, 3, 5	100
4	79	2, 5	59	2, 4, 5	100
5	59	3, 4	79	3, 4, 5	100
		3, 5	100		
		4, 5	100		

Insight 3 *Selecting the optimal revision stages is essential. Revising decisions at the optimal stages can outperform revising decisions more often in arbitrary stages.*

Table 4 reports the VAMS for each possible combination of revision points, and indicates the best ones in bold. For $\mu = 1$, we observe that fixing the revision point to the 4th stage yields the highest VAMS value while the VAMS in the 3rd stage is only marginally lower than the optimal value. However, the VAMS significantly decreases when revisions are made in the 2nd or 5th stages. When two revisions are allowed, the best results are achieved with revision points (3,5) and (4,5). Since the 4th and 3rd periods are the most effective ones in the one revision case, the optimal combination for the two-revision case can be anticipated as (3,4). However, our analysis indicates that this is not the case, highlighting the complex nature of revision decisions. The results show the importance of not only the number but also the timing of revisions. For instance, revising twice in (3,5) or (4,5) is more effective than three times in (2,3,4). Similarly, revising once in 3 or 4 is more effective than twice in (2,5). Overall, the result of the adaptive method is highly impacted by the timing of revision, emphasizing the significance of optimally choosing the revision points in situations with limited flexibility.

5.2. Adaptive Multistage Stochastic Generation Expansion Planning Problem

We study a generation expansion planning (GEP) problem: a process used by energy system planners to determine the optimal mix of generation sources to meet future power demand. This problem is commonly tackled using multistage stochastic programming methods, where the goal is to primarily adjust the generation expansion decisions (i.e., the installed capacity of energy generation sources) and consequently the operational decisions (i.e., how much power to produce) as uncertainties realize over a multi-period planning horizon. However, fully flexible policies obtained by multistage stochastic programming are not practically feasible for expansion decisions due to restrictions such as contractual commitments, and lead time requirements (Munoz et al. 2013, Careri et al. 2011). In such cases where flexibility is limited, an effective solution is selecting critical stages for revising decisions while committing to these decisions in between. Thus, we introduce an AMSP approach to solve the GEP problem. We consider the allowed – practically deemed feasible – flexibility level of the underlying GEP process. Consequently, expansion decisions will only be revised at the most critical stages determined by our optimization model, leading to more practically efficient planning outcomes.

We consider a scenario tree \mathcal{T} with a finite planning horizon $\mathcal{P} = \{1, \dots, T\}$. The probability of uncertainty realization at node $n \in \mathcal{T}$ is denoted by p_n . Each stage t is divided into subperiods \mathcal{K}_t reflecting hours within that stage. Then, the total number of hours at node $n \in \mathcal{T}$ in subperiod $k \in \mathcal{K}_{t_n}$ is denoted as h_{nk} . We consider a set of different generators denoted by $i \in \mathcal{G}$. The initial number of generators of type $i \in \mathcal{I}$ at the beginning of the planning is denoted as x_i^0 . The construction of a generator of type $i \in \mathcal{I}$ at node $n \in \mathcal{T}$ incurs a capital cost of c_{in}^c , a fixed operations and maintenance (O&M) cost of c_{in}^f , and a variable O&M cost of c_{in}^v . The maximum number of generators of type $i \in \mathcal{I}$ that can be constructed at node $n \in \mathcal{T}$ is limited to x_i^{\max} . Each generator type $i \in \mathcal{I}$ has a maximum generation capacity of m_i^{\max} . Additionally, each generator i in \mathcal{I} has a capacity factor l_{ink} , which indicates the amount of energy it can produce at node $n \in \mathcal{T}$ during subperiod $k \in \mathcal{K}_{t_n}$, relative to its maximum generation capacity. The fuel price of a generator of type $i \in \mathcal{I}$ at node $n \in \mathcal{T}$ is denoted as c_{in}^u . The hourly demand at node $n \in \mathcal{T}$ in subperiod $k \in \mathcal{K}_{t_n}$ is given by d_{nk} . Failure to meet the demand incurs a penalty cost of c^p . The yearly interest rate is represented by γ . Note that capacity factor l_{ink} and demand d_{nk} are stochastic parameters.

Let x_{in} be an integer decision variable that indicates the number of type $i \in \mathcal{I}$ generators constructed at node $n \in \mathcal{T}$. Let y_{ink} be a continuous decision variable for the generation of a type $i \in \mathcal{I}$ generator in subperiod $k \in \mathcal{K}_{t_n}$ at node $n \in \mathcal{T}$. Finally, let u_{nk} be a continuous variable for unsupplied demand in subperiod $k \in \mathcal{K}_{t_n}$ at node $n \in \mathcal{T}$. We then formulate the adaptive multistage GEP based on the formulation in Zou et al. (2018), as follows:

$$\min \sum_{n \in \mathcal{T}} p_n \frac{1}{(1 + \gamma)^{t_n - 1}} \left(\sum_{i \in \mathcal{G}} \left(c_{in}^c + \sum_{t=t_n}^T \frac{c_{in}^f}{(1 + \gamma)^{t - t_n}} \right) m_i^{\max} x_{in} \right)$$

$$\begin{aligned}
& + \sum_{i \in \mathcal{G}} \sum_{k \in \mathcal{K}_{t_n}} (c_{in}^v + c_{in}^u) h_{nk} y_{ink} + \sum_{k \in \mathcal{K}_{t_n}} c^p h_{nk} u_{nk} \Big) \quad (17a) \\
\text{s.t. } & \sum_{m \in P(n)} x_{im} + x_i^0 \geq y_{ink} \quad \forall i \in \mathcal{G}, k \in \mathcal{K}_{t_n}, n \in \mathcal{T} \quad (17b) \\
& x_{in} \leq x_{in}^{\max} \quad \forall i \in \mathcal{G}, n \in \mathcal{T} \quad (17c) \\
& \sum_{i \in \mathcal{G}} l_i y_{ink} + u_{nk} \geq d_{nk} \quad \forall k \in \mathcal{K}_{t_n}, n \in \mathcal{T} \quad (17d) \\
& x_{in} \in \mathbb{Z}_+, y_{ink} \in \mathbb{R}_+ \quad \forall i \in \mathcal{G}, k \in \mathcal{K}_{t_n}, n \in \mathcal{T} \quad (17e) \\
& (4d) - (4g), (8)
\end{aligned}$$

The objective function (17a) minimizes the expected cost of generator construction, fixed and variable operations and maintenance, fuel prices, and penalty for unsupplied demand over a planning period, discounting it to the beginning of the planning period. Constraint (17b) guarantees that the generation amount does not exceed the total generation capacity of constructed generators. Constraint (17c) limits the number of generators that can be constructed. Constraint (17d) ensures demand is satisfied. Finally, constraints (4d) - (4g), (8) are the necessary NACs.

5.2.1. Experimental Setup. We set model parameters based on real-world data and optimize the yearly investment choices for various energy sources, ensuring each source operates within its capacity limits while meeting the system's demand for each sub-period. We consider 2020 as the base year. We include five different types of generators: a traditional gas-powered combined-cycle generator, a gas-powered combined-cycle generator with carbon capture (CC), and three renewable energy sources: onshore wind, offshore wind, and solar PV. We set the capital and O&M cost parameters based on the ‘‘Capital Cost and Performance Characteristic Estimates for Utility Scale Electric Power Generating Technologies’’ report from the Energy Information Administration (EIA) (The U.S. Energy Information Administration 2020). The report provides various characteristics of these technologies, such as their net nominal capacity, capital cost, and fixed and variable O&M costs. We set the cost parameters at the base year (c_{i0}^c , c_{i0}^f , c_{i0}^v) according to values provided in Table 5. By utilizing the ‘‘EIA Electric Power Annual 2021’’ (The U.S. Energy Information Administration 2022), we set the natural gas price for the combined-cycle power plants, denoted as c_{in}^u , to 0.04\$ per kWh. Based on Basciftci et al. (2019), we assume a yearly decrease of 10% in capital costs for renewable energy sources and 5% for the combined-cycle with CC, reflecting expected technology development. Additionally, we consider a 10% yearly increase in variable O&M costs and fuel prices for both the combined-cycle and the combined-cycle with CC. Other cost parameters remain constant throughout the planning horizon.

Following the study by Min et al. (2018), we assume that each year has four subperiods differentiated by their demand level: base, off-peak, shoulder, and peak. The demand per subperiod (d_{nk})

Table 5 Generator costs in the baseline year 2020

Type of Generator	Nominal Capacity (kW)	Capital (\$/kW)	Fixed O&M (\$/kW-year)	Variable O&M (\$/MWh)
Combined-Cycle	418	1084.0	14.1	2.6
Combined-Cycle with 90% CC	377	2481.0	27.6	5.8
Onshore Wind	200	1265.0	26.4	0
Offshore Wind	400	4375.0	110.0	0
Solar PV	150	1313.0	15.3	0

is calculated by multiplying the node's demand (d_n) by the corresponding weights (w_k) of 0.9, 1.1, 1.3, and 1.5, respectively, representing the base, off-peak, shoulder, and peak subperiods. The subperiods have different durations within a year, with the base period covering 55%, the off-peak period covering 40%, the shoulder period covering 4.95%, and the peak period covering 0.05%. Based on Singh et al. (2009), the scenario tree is generated to capture the stochastic nature of the demand and capacity factor for renewable energy sources as in Algorithm 2 in Appendix B.

We assume that demand at the root node d_1 is 1000MWh. Each node n of the scenario tree, except the root node, has a randomly allocated growth factor λ_{nk}^d for all subperiods $k \in \mathcal{K}_{tn}$. These growth rates are derived from a normal distribution with a mean of 5% (μ^d) and a standard deviation of 5% (σ^d), reflecting the yearly increase in global electricity demand over recent years as reported in International Energy Agency (2023). The amount of energy produced by renewable energy sources is determined by their capacity factor l_{ink} . To reflect the uncertainty of renewable energy production, we generate random samples of capacity factors from a normal distribution with $\mathcal{N}(30\%, 10\%)$ for onshore wind, $\mathcal{N}(60\%, 5\%)$ for offshore wind, and $\mathcal{N}(20\%, 10\%)$ for solar PV. We assume the network consists of only combined-cycle generators in the base year. Accordingly, we adjust the initial capacities x_i^0 to ensure that they can cover the average demand if operated at full capacity. We set the yearly interest rate γ to 5%.

5.2.2. Optimal Revision Points. Table 6 shows the optimal decisions for the revision points and VAMS for each type of generator across different values of μ ranging from 1 to 4 when $T = 9$.

Table 6 Optimal revision points for the GEP with $T = 9$

Type of Generator	Number of revisions			
	$\mu=1$	$\mu=2$	$\mu=3$	$\mu=4$
Combined-Cycle	4	4, 7	2, 4, 6	2, 4, 6, 7
Combined-Cycle with 90% CC	9	6, 8	5, 6, 9	5, 7, 8, 9
Onshore Wind	5	4, 7	4, 6, 8	4, 6, 8, 9
Offshore Wind	7	6, 8	6, 8, 9	4, 6, 8, 9
Solar PV	9	7, 9	7, 8, 9	5, 7, 8, 9
VAMS (%)	76.7	88.3	92.9	95.7

Insight 4 *Optimal revision points for the traditional combined-cycle gas-powered generator are relatively early compared to the revision points for renewable energy sources and the combined-cycle gas-powered generator with CC. This shows the importance of allowing each generator to be revised at different stages, as advocated by AMSP.*

The structure of optimal revision points can be explained by the underlying dynamics of the GEP. Capacity expansion decisions for combined-cycle generators occur in earlier periods due to increasing O&M costs and natural gas prices over the years. Consequently, optimal revision points for combined-cycle generators are at earlier periods. However, as capital costs of the combined-cycle with 90% CC decrease over time due to advancements in CC technologies, expansions are shifted to later periods. Likewise, renewable capacities also expand during these later periods. The main reason is the declining capital costs of renewables and the increased variability of their capacity factors within the scenario tree. As a result, capacity expansion amounts during these later periods show notable differences depending on the realization of uncertainties. Therefore, optimal revision points for wind and solar are at later periods. To summarize, optimizing power generation expansion planning requires determining tailored revision points for each generator type, considering evolving costs, technological advancements, and market dynamics.

The results indicate a positive correlation between the number of revisions and VAMS. Moreover, as the number of revisions increases, we observe a diminishing return, which aligns with Insight 1. Additionally, our findings support Insight 2, as multiple revisions yield significantly higher VAMS values than a single revision.

5.2.3. Computational Performance. We analyze the impact of our NACs elimination techniques and our solution algorithm on the computational performance of AMSP. To this end, we first replace the NACs in Model (17) (corresponding to formulation $(AMS_{\mu+})$) with NACs in formulation (AMS_{μ}) . Accordingly, we obtain model $\{(17a) : (17b) - (17e), (4b) - (4g)\}$, and we call it **GEP** which corresponds to the original AMSP formulation without the NACs elimination techniques. We call the improved model, Model (17), as **GEP+**.

We present the computational results of **GEP**, **GEP+**, and Algorithm 1 in Table 7, which includes solution times, optimality gaps, and VAMS for each approach under various instances. These results are under a threshold of 0.1% optimality gap and a time limit of 5 hours. For our analysis, we generate instances with varying flexibility levels (μ ranging from 1 to 4) and numbers of stages (T ranging from 5 to 9). The scenario tree utilized in our experiments consists of 3 branches per node.

We observe that **GEP** is outperformed by **GEP+** in all instances. **GEP** gives memory error failing to solve any instance with more than eight stages and does not provide feasible solutions within the time limit. Moreover, the instance with seven stages and one revision cannot be solved. On

Table 7 Performance of GEP, GEP+ and Algorithm 1

Stage	μ	GEP		GEP+		Algorithm 1	
		Time (s)	VAMS (%)	Time (s)	VAMS (%)	Time (s)	VAMS (%)
5	1	5.7	75.2	4.1	75.2	0.7	75.2
	2	1.7	88.2	0.8	88.2	0.5	88.2
	3	0.6	97.7	0.5	97.7	0.2	97.7
6	1	188.6	78.0	6.1	78.0	3.6	78.0
	2	16.4	90.1	8.2	90.1	4.3	90.1
	3	10.6	95.3	6.6	95.3	2.8	95.3
	4	7.8	98.8	2.4	98.8	2.4	98.8
7	1	-	-	70.6	78.4	21.4	78.4
	2	389.5	90.7	67.0	90.7	14.6	90.7
	3	113.7	94.4	26.7	94.4	11.5	94.4
	4	92.3	98.2	18.7	98.2	8.4	98.2
8	1	-	-	692.7	77.9	203.1	77.9
	2	-	-	264.5	88.9	200.0	88.9
	3	-	-	136.2	94.3	86.1	94.3
	4	-	-	88.8	96.7	50.5	96.7
9	1	-	-	18000.0	76.5	1061.6	76.7
	2	-	-	12605.7	88.3	2573.5	88.3
	3	-	-	2552.7	92.9	1596.8	92.9
	4	-	-	1395.6	95.7	433.7	95.7
Average		-	-	1892.0	89.3	331.7	89.3

the other hand, **GEP+** can solve all but one instance to optimality. The optimality gap of that single instance (with nine stages and one revision) is 0.5%. The results show that including the elimination techniques of the NACs leads to an average decrease in computation time of 59%, with up to 97%. We also observe that as the size of the problem increases, the superiority of **GEP+** over **GEP** in terms of solution time becomes more prominent. This is due to the significant increase in the number NACs for larger instances as reported in Table 3.

We show that the solution time can be further reduced with Algorithm 1. Compared to **GEP**, the solution time is decreased on average by 82% with up to 98%. Compared to **GEP+**, solution time is decreased on average 56% with up to 94%. It is worth noting that although Algorithm 1 includes heuristic steps, it can still attain the optimal value for all instances. Overall, Algorithm 1 proves to be highly effective in reducing solution times and delivering optimal solutions for AMSPs.

6. Conclusions

In this paper, we propose a new optimization approach called AMSP to achieve the optimal balance between decision flexibility and commitment by deciding on the best times to revise decisions under uncertainty. We present a generic mathematical formulation for the proposed approach and prove the NP-Hardness of the proposed problem. Then, we derive theoretical properties for elimination of NACs as their existence increases the problem complexity. We provide a tailored solution algorithm for solving the generic class of AMSPs with mixed-integer master problem and subproblems, by

combining L-shaped and Benders decomposition methods. To further accelerate the algorithm, we include a heuristic-based preprocessing step that can eliminate a set of candidate feasible solutions.

Computational experiments on lot-sizing and generation expansion planning demonstrate the significant benefits of optimally choosing revision moments when flexibility is limited. By making revisions only at these optimal moments, businesses can produce results comparable to a fully flexible setting while respecting the flexibility limitations of the decision-making process. Remarkably, in some cases, we show that the optimal timing of revisions can even outperform frequency. Next to these practical insights, we demonstrate that eliminating NACs improves computational performance over various instances by an average of 59%, with our solution algorithm providing a further 56% enhancement on average.

Acknowledgements

This project has received funding from the Fuel Cells and Hydrogen 2 Joint Undertaking (now Clean Hydrogen Partnership) under Grant Agreement No 875090. This Joint Undertaking receives support from the European Union’s Horizon 2020 research and innovation programme, Hydrogen Europe and Hydrogen Europe Research. Albert H. Schrottenboer is supported by a VENI research talent grant from the Dutch Science Foundation (NWO).

References

- Aminov, I., De Smet, A., Jost, G., and Mendelsohn, D. (2019). Decision making in the age of urgency. *McKinsey & Company*.
- Angulo, G., Ahmed, S., and Dey, S. S. (2016). Improving the integer L-shaped method. *INFORMS Journal on Computing*, 28(3):483–499.
- Apap, R. M. and Grossmann, I. E. (2017). Models and computational strategies for multistage stochastic programming under endogenous and exogenous uncertainties. *Computers & Chemical Engineering*, 103:233–274.
- Basciftci, B., Ahmed, S., and Gebraeel, N. (2019). Adaptive two-stage stochastic programming with an application to capacity expansion planning. *arXiv preprint arXiv:1906.03513*.
- Basciftci, B., Ahmed, S., and Shen, S. (2021). Distributionally robust facility location problem under decision-dependent stochastic demand. *European Journal of Operational Research*, 292(2):548–561.
- Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252.
- Bertsimas, D., Shtern, S., and Sturt, B. (2023). A data-driven approach to multistage stochastic linear optimization. *Management Science*, 69(1):51–74.
- Birge, J. R. (1985). Decomposition and partitioning methods for multistage stochastic linear programs. *Operations Research*, 33(5):989–1007.

- Birge, J. R. and Louveaux, F. (2000). *Introduction to Stochastic Programming*. Springer New York.
- Boland, N., Dumitrescu, I., Froyland, G., and Kalinowski, T. (2016). Minimum cardinality non-anticipativity constraint sets for multistage stochastic programming. *Mathematical Programming*, 157(1):69–93.
- Careri, F., Genesi, C., Marannino, P., Montagna, M., Rossi, S., and Siviero, I. (2011). Generation expansion planning in the age of green economy. *IEEE Transactions on Power Systems*, 26(4):2214–2223.
- Chemama, J., Cohen, M. C., Lobel, R., and Perakis, G. (2019). Consumer subsidies with a strategic supplier: Commitment vs. flexibility. *Management Science*, 65(2):681–713.
- Colvin, M. and Maravelias, C. T. (2010). Modeling methods and a branch and cut algorithm for pharmaceutical clinical trial planning using stochastic programming. *European Journal of Operational Research*, 203(1):205–215.
- Daryalal, M., Bodur, M., and Luedtke, J. R. (2022). Lagrangian dual decision rules for multistage stochastic mixed-integer programming. *Operations Research*.
- Drusvyatskiy, D. and Xiao, L. (2022). Stochastic optimization with decision-dependent distributions. *Mathematics of Operations Research*.
- Fischetti, M., Ljubić, I., and Sinnl, M. (2017). Redesigning benders decomposition for large-scale facility location. *Management Science*, 63(7):2146–2162.
- Gallo, A. (2010). In a volatile world, your strategy must be flexible. *Harvard Business Review*.
- Garey, M. and Johnson, D. (1980). *Computers and Intractability. A Guide to the Theory of NP-Completeness*. Freeman, New York.
- Geoffrion, A. M. (1972). Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260.
- Goel, V. and Grossmann, I. E. (2006). A class of stochastic programs with decision dependent uncertainty. *Mathematical Programming*, 108(2):355–394.
- Guan, Y. and Miller, A. J. (2008). Polynomial-time algorithms for stochastic uncapacitated lot-sizing problems. *Operations Research*, 56(5):1172–1183.
- Gul, S., Denton, B. T., and Fowler, J. W. (2015). A progressive hedging approach for surgery planning under uncertainty. *INFORMS Journal on Computing*, 27(4):755–772.
- Gupta, A., Nagarajan, V., and Singla, S. (2016). Algorithms and adaptivity gaps for stochastic probing. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 1731–1747. SIAM.
- Gupta, V. and Grossmann, I. E. (2011). Solution strategies for multistage stochastic programming with endogenous uncertainties. *Computers & Chemical Engineering*, 35(11):2235–2247.
- Hellemo, L., Barton, P. I., and Tomasgard, A. (2018). Decision-dependent probabilities in stochastic programs with recourse. *Computational Management Science*, 15(3):369–395.

- Herrera, C., Belmokhtar-Berraf, S., Thomas, A., and Parada, V. (2016). A reactive decision-making approach to reduce instability in a master production schedule. *International Journal of Production Research*, 54(8):2394–2404.
- Huang, K. and Ahmed, S. (2009). The value of multistage stochastic programming in capacity planning under uncertainty. *Operations Research*, 57(4):893–904.
- International Energy Agency (2023). Electricity Market Report.
- Köhler, J., Geels, F. W., Kern, F., Markard, J., Onsongo, E., Wieczorek, A., Alkemade, F., Avelino, F., Bergek, A., Boons, F., et al. (2019). An agenda for sustainability transitions research: State of the art and future directions. *Environmental Innovation and Societal Transitions*, 31:1–32.
- Kouvelis, P., Chambers, C., and Wang, H. (2006). Supply chain management research and production and operations management: Review, trends, and opportunities. *Production and Operations Management*, 15(3):449–469.
- Laporte, G. and Louveaux, F. V. (1993). The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13(3):133–142.
- Latorre, G., Cruz, R. D., Areiza, J. M., and Villegas, A. (2003). Classification of publications and models on transmission expansion planning. *IEEE Transactions on Power Systems*, 18(2):938–946.
- Mankins, M. (2022a). In a volatile world, your strategy must be flexible. *Harvard Business Review*.
- Mankins, M. (2022b). In uncertain times, the best strategy is adaptability. *Harvard Business Review*.
- Meadowcroft, J. (2009). What about the politics? sustainable development, transition management, and long term energy transitions. *Policy Sciences*, 42:323–340.
- Meistering, M. and Stadtler, H. (2017). Stabilized-cycle strategy for capacitated lot sizing with multiple products: Fill-rate constraints in rolling schedules. *Production and Operations Management*, 26(12):2247–2265.
- Milner, J. M. and Kouvelis, P. (2005). Order quantity and timing flexibility in supply chains: The role of demand characteristics. *Management Science*, 51(6):970–985.
- Min, D., Ryu, J.-h., and Choi, D. G. (2018). A long-term capacity expansion planning model for an electric power system integrating large-size renewable energy technologies. *Computers & Operations Research*, 96:244–255.
- Munoz, F. D., Hobbs, B. F., Ho, J. L., and Kasina, S. (2013). An engineering-economic approach to transmission planning under market and regulatory uncertainties: Wecc case study. *IEEE Transactions on Power Systems*, 29(1):307–317.
- New Energy Coalition (2020). The Northern Netherlands hydrogen investment plan 2020: Expanding the Northern Netherlands hydrogen valley. Technical report, New Energy Coalition.
- Pan, K. and Guan, Y. (2016). Strong formulations for multistage stochastic self-scheduling unit commitment. *Operations Research*, 64(6):1482–1498.

- Powell, W. B. (2019). A unified framework for stochastic optimization. *European Journal of Operational Research*, 275(3):795–821.
- Rahmaniani, R., Ahmed, S., Crainic, T. G., Gendreau, M., and Rei, W. (2020). The benders dual decomposition method. *Operations Research*, 68(3):878–895.
- Rudloff, B., Street, A., and Valladão, D. M. (2014). Time consistency and risk averse dynamic decision models: Definition, interpretation and practical consequences. *European Journal of Operational Research*, 234(3):743–750.
- Ruszczynski, A. P. and Shapiro, A. (2003). *Stochastic Programming*. Amsterdam: Elsevier.
- Şafak, Ö., Çavuş, Ö., and Aktürk, M. S. (2022). A two-stage decision dependent stochastic approach for airline flight network expansion. *Transportation Research Part B: Methodological*, 158:78–101.
- Shapiro, A., Dentcheva, D., and Ruszczyński, A. P. (2009). *Lectures on Stochastic Programming: Modeling and Theory*. Society for Industrial and Applied Mathematics.
- Shapiro, A. and Nemirovski, A. (2005). On complexity of stochastic programming problems. *Continuous Optimization: Current Trends and Modern Applications*, pages 111–146.
- Singh, K. J., Philpott, A. B., and Wood, R. K. (2009). Dantzig-wolfe decomposition for solving multistage stochastic capacity-planning problems. *Operations Research*, 57(5):1271–1286.
- Singla, S. (2018). Combinatorial optimization under uncertainty: Probing and stopping-time algorithms. *Unpublished doctoral dissertation, Carnegie Mellon University*.
- The U.S. Energy Information Administration (2020). Capital cost and performance characteristic estimates for utility scale electric power generating technologies.
- The U.S. Energy Information Administration (2022). Electric Power Annual.
- Van Slyke, R. M. and Wets, R. (1969). L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4):638–663.
- Wadhwa, S., Saxena, A., and Chan, F. T. S. (2008). Framework for flexibility in dynamic supply chain management. *International Journal of Production Research*, 46(6):1373–1404.
- Zou, J., Ahmed, S., and Sun, X. A. (2018). Partially adaptive stochastic optimization for electric power generation expansion planning. *INFORMS Journal on Computing*, 30(2):388–401.

Appendix A: Proofs of Lemmas and Propositions

This section presents the proofs of the lemmas and propositions discussed in the main paper. To enhance readability, corresponding statements are restated below.

A.1. Proof of Lemma 1

Lemma 1 For (AMS_μ) , there exist an optimal solution where $r_{iT} \leq \mu$ is binding for all $i \in \mathcal{I}$.

Proof: Assume that constraint $r_{iT}^* \leq \mu$ has a slack $\epsilon \in \{1, \dots, \mu\}$ for at least one $i \in \mathcal{I}$ at an optimal solution \mathbf{r}^* . Let $\mathcal{I}' \subseteq \mathcal{I}$ be the set of indices of the state variables with slack at this solution. Since any such solution under \mathbf{r}^* has an objective function value which is same as the solutions we can construct by imposing $r_{iT} = \mu$ for all $i \in \mathcal{I}'$, the desired result follows as we can obtain a different but still optimal solution where $r_{iT} = \mu$ for all $i \in \mathcal{I}$. \square

A.2. Proof of Proposition 1

Proposition 1 *For $\mu = 0$, formulation (AMS_μ) is equivalent to two-stage stochastic programming formulation $(2SP)$. For $\mu = T - 1$, formulation (AMS_μ) is equivalent to multistage stochastic programming formulation (MSP) .*

Proof: If $\mu = 0$, then $r_{it} = 0$ for all $i = 1, \dots, I, t = 1, \dots, T$. Then (4b) and (4c) reduce to $x_{im} = x_{in}$, which is analogous to constraint (2b) of $(2SP)$. On the other hand, for $\mu = T - 1$, there exist an optimal solution where $r_{i(t+1)} = r_{it} + 1$ with $r_{iT} = T - 1$ for $i = 1, \dots, I, t = 1, \dots, T - 1$ based on Lemma 1. As a result, constraints (4b) and (4c) become redundant. Thus, it can be concluded that $(AMS_{(T-1)})$ is equivalent to (MSP) . \square

A.3. Proof of Proposition 2

Proposition 2 $z(2SP) = z(AMS_0) \geq z(AMS_1) \geq \dots \geq z(AMS_{(T-2)}) \geq z(AMS_{(T-1)}) = z(MSP)$.

Proof: To demonstrate this result, we show that the feasible region of $(AMS_{(\mu-1)})$ is contained in the feasible region of (AMS_μ) for any $\mu \in \{1, \dots, T - 1\}$. Note the constraint sets of the problems $(AMS_{(\mu-1)})$ and (AMS_μ) are the same except constraint (4d), where the upper bound on the r_{it} variables are affected by the change in μ value for every $i \in \mathcal{I}$ and $t \in \mathcal{P}$. While $AMS_{(\mu-1)}$ has $r_{iT} \leq \mu - 1$, AMS_μ has $r_{iT} \leq \mu$ where $r_{iT} \in \mathbb{Z}_+$. Since $\mu - 1 \leq \mu$, $z(AMS_{(\mu-1)}) \geq z(AMS_\mu)$ for any $\mu \in \{1, \dots, T - 1\}$. \square

A.4. Proof of Proposition 3

Proposition 3 *AMSP is NP-Hard.*

Proof: To prove the Proposition 3, we demonstrate that the feasibility problem associated with the adaptive multistage problem (3) is \mathcal{NP} -Complete. To this end, we construct a feasibility problem, referred as P, whose feasible region is defined in (18). Then, we show that the subset sum problem, which is known to be \mathcal{NP} -Complete (Garey and Johnson 1980), can be reduced to P in polynomial time. We specify the subset sum problem as follows: Given the non-negative integers $\{w'_{il}\}_{i \in \mathcal{I}', l \in \mathcal{L}}$ and W' , does there exist $\mathcal{H}_l \subseteq \mathcal{I}'$ for all $l \in \mathcal{L}$ such that $\sum_{l \in \mathcal{L}} \sum_{i \in \mathcal{H}_l} w'_{il} = W'$?

$$\min_{\alpha, \beta, \mathbf{t}^*} 0 \tag{18a}$$

$$\text{s.t. } \alpha_{i2} - \beta_{i2} = 0 \tag{18b} \quad i \in \mathcal{I}$$

$$\alpha_{i3} - \beta_{i3} = \alpha_{i1} \tag{18c} \quad i \in \mathcal{I}$$

$$\alpha_{i4} - \beta_{i4} = 0 \tag{18d} \quad i \in \mathcal{I}$$

$$\alpha_{i5} - \beta_{i5} = \alpha_{i2} \tag{18e} \quad i \in \mathcal{I}$$

$$\alpha_{i6} - \beta_{i6} = 0 \tag{18f} \quad i \in \mathcal{I}$$

$$\alpha_{i7} - \beta_{i7} = \alpha_{i2} \quad i \in \mathcal{I} \quad (18g)$$

$$\alpha_{i8} - \beta_{i8} = 0 \quad i \in \mathcal{I} \quad (18h)$$

$$\alpha_{i9} - \beta_{i9} = \alpha_{i4} \quad i \in \mathcal{I} \quad (18i)$$

$$\alpha_{i10} - \beta_{i10} = 0 \quad i \in \mathcal{I} \quad (18j)$$

$$\alpha_{i11} - \beta_{i11} = \alpha_{i4} \quad i \in \mathcal{I} \quad (18k)$$

$$\alpha_{i12} - \beta_{i12} = 0 \quad i \in \mathcal{I} \quad (18l)$$

$$\alpha_{i13} - \beta_{i13} = \alpha_{i4} \quad i \in \mathcal{I} \quad (18m)$$

$$\alpha_{i14} - \beta_{i14} = 0 \quad i \in \mathcal{I} \quad (18n)$$

$$\alpha_{i15} - \beta_{i15} = \alpha_{i4} \quad i \in \mathcal{I} \quad (18o)$$

$$\sum_{i \in \mathcal{I}} (w_{i1}\alpha_{i1} + w_{i2}\alpha_{i2} + w_{i4}\alpha_{i4}) = W \quad (18p)$$

$$\alpha_{im} = \alpha_{in}, \beta_{im} = \beta_{in} \quad \forall m, n \in S_t, \quad t < t_{i1}^* \quad i \in \mathcal{I} \quad (18q)$$

$$\alpha_{im} = \alpha_{in}, \beta_{im} = \beta_{in} \quad \forall m, n \in S_t \cap \mathcal{T}(l, t_{ik+1}^*), \quad l \in S_{t_{ik}^*}, \quad t_{ik+1}^* > t \geq t_{ik}^* \quad i \in \mathcal{I} \quad (18r)$$

$$\alpha_{im} = \alpha_{in}, \beta_{im} = \beta_{in} \quad \forall m, n \in S_t \cap \mathcal{T}(l), \quad l \in S_{t_{i\mu}^*}, \quad t \geq t_{i\mu}^* \quad i \in \mathcal{I} \quad (18s)$$

$$t_{ik}^* \in \Gamma_i^* \subseteq \{2, \dots, T\} \quad i \in \mathcal{I}, k \in \{1, \dots, \mu\} \quad (18t)$$

First observe that the problem (18) is an instance of the adaptive multistage problem by letting $\mathcal{J} = \emptyset$, and $x_{in} = \begin{bmatrix} \alpha_{in} \\ \beta_{in} \end{bmatrix}$. We consider the scenario tree \mathcal{T} with $T = 4$ stages and $B = 2$ branches. Each stage of the scenario tree consists of the following nodes: $S_1 = \{1\}$, $S_2 = \{2, 3\}$, $S_3 = \{4, 5, 6, 7\}$, $S_4 = \{8, 9, 10, 11, 12, 13, 14, 15\}$. We let $\mathcal{I} = \mathcal{I}'$, $W = W'$, and $w_{il} = w'_{il}$ for all $i \in \mathcal{I}, l \in \mathcal{L} = \{1, 2, 4\}$. We note that constraints (18b)–(18p) correspond to constraint (1b). Similarly, constraints (18q)–(18t) refer to the constraints (3b)–(3e), respectively. In Lemma 2, we list all possible combinations of revision points for varying μ from zero to three. We note that the first case, where $\Gamma_i^* = \{\}$, corresponds to a scenario with no revisions during planning. From the second to the fourth case, there is one revision, from the fifth to the seventh case, there are two revisions, and the last case is fully flexible.

Lemma 2 *Under the given set of revision points Γ_i^* for all $i \in \mathcal{I}$, we can construct the following feasible solutions for the problem (18):*

1. If $\Gamma_i^* = \{\}$, then $\alpha_{i1} = \alpha_{i2} = \alpha_{i4} = 0$ for all $i \in \mathcal{I}$.
2. If $\Gamma_i^* = \{2\}$, then $\alpha_{i1} = 1, \alpha_{i2} = \alpha_{i4} = 0$ for all $i \in \mathcal{I}$.
3. If $\Gamma_i^* = \{3\}$, then $\alpha_{i2} = 1, \alpha_{i1} = \alpha_{i4} = 0$ for all $i \in \mathcal{I}$.
4. If $\Gamma_i^* = \{4\}$, then $\alpha_{i4} = 1, \alpha_{i1} = \alpha_{i2} = 0$ for all $i \in \mathcal{I}$.
5. If $\Gamma_i^* = \{2, 3\}$, then $\alpha_{i1} = \alpha_{i2} = 1, \alpha_{i4} = 0$ for all $i \in \mathcal{I}$.
6. If $\Gamma_i^* = \{2, 4\}$, then $\alpha_{i1} = \alpha_{i4} = 1, \alpha_{i2} = 0$ for all $i \in \mathcal{I}$.
7. If $\Gamma_i^* = \{3, 4\}$, then $\alpha_{i2} = \alpha_{i4} = 1, \alpha_{i1} = 0$ for all $i \in \mathcal{I}$.
8. If $\Gamma_i^* = \{2, 3, 4\}$, then $\alpha_{i1} = \alpha_{i2} = \alpha_{i4} = 1$ for all $i \in \mathcal{I}$.

Given a solution $\{\mathcal{H}_l\}_{l \in \mathcal{L}}$ to the subset-sum problem, we can construct a feasible solution for problem (18) based on Lemma 2. For all $l \in \{1, 2, 4\}$, we set $\alpha_{il} = 1$ if $i \in \mathcal{H}_l$ and 0 otherwise. For example, in the first case: if $\alpha_{i1} = \alpha_{i2} = \alpha_{i4} = 0$, then $\alpha_{in} = \beta_{in} = 0$ for all $i \in \mathcal{I}, n \in \mathcal{T}$. The second case: $\alpha_{i1} = 1, \alpha_{i2} = \alpha_{i4} = 0$, then $\alpha_{i3} = 1, \alpha_{in} = 0$ for all $i \in \mathcal{I}, n \in \mathcal{T} \setminus \{1, 3\}$ and $\beta_{in} = 0$ for all $i \in \mathcal{I}, n \in \mathcal{T}$ where $t_{i1}^* = 2$. The third case: $\alpha_{i2} = 1, \alpha_{i1} = \alpha_{i4} = 0$, then $\alpha_{i3} = \alpha_{i5} = \alpha_{i7} = 1, \alpha_{in} = 0$ for all $i \in \mathcal{I}, n \in \mathcal{T} \setminus \{2, 3, 5, 7\}$ and $\beta_{i2} = \beta_{i3} = 1, \beta_{in} = 0$ for all $i \in \mathcal{I}, n \in \mathcal{T} \setminus \{2, 3\}$ where $t_{i1}^* = 3$. The fourth case: $\alpha_{i4} = 1, \alpha_{i1} = \alpha_{i2} = 0$, then $\alpha_{i5} = \alpha_{i6} = \alpha_{i7} = \alpha_{i9} = \alpha_{i11} = \alpha_{i13} = \alpha_{i15} = 1, \alpha_{in} = 0$ for all $i \in \mathcal{I}, n \in \mathcal{T} \setminus \{4, 5, 6, 7, 9, 11, 13, 15\}$ and $\beta_{i4} = \beta_{i5} = \beta_{i6} = \beta_{i7} = 1, \beta_{in} = 0$ for all $i \in \mathcal{I}, n \in \mathcal{T} \setminus \{4, 5, 6, 7\}$ where $t_{i1}^* = 4$. The fifth case: $\alpha_{i1} = \alpha_{i2} = 1, \alpha_{i4} = 0$, then $\alpha_{i3} = 2, \alpha_{i5} = \alpha_{i7} = 1, \alpha_{in} = 0$ for all $i \in \mathcal{I}, n \in \mathcal{T} \setminus \{1, 2, 3, 5, 7\}$ and $\beta_{i2} = \beta_{i3} = 1, \beta_{in} = 0$ for all $i \in \mathcal{I}, n \in \mathcal{T} \setminus \{2, 3\}$ where $t_{i1}^* = 2, t_{i2}^* = 3$. The sixth case: $\alpha_{i1} = \alpha_{i4} = 1, \alpha_{i2} = 0$, then $\alpha_{i3} = \alpha_{i5} = \alpha_{i6} = \alpha_{i7} = \alpha_{i9} = \alpha_{i11} = \alpha_{i13} = \alpha_{i15} = 1, \alpha_{in} = 0$ for all $i \in \mathcal{I}, n \in \mathcal{T} \setminus \{1, 3, 4, 5, 6, 7, 9, 11, 13, 15\}$ and $\beta_{i4} = \beta_{i5} = \beta_{i6} = \beta_{i7} = 1, \beta_{in} = 0$ for all $i \in \mathcal{I}, n \in \mathcal{T}$ where $t_{i1}^* = 2, t_{i2}^* = 4$. The seventh case: $\alpha_{i2} = \alpha_{i4} = 1, \alpha_{i1} = 0$, then $\alpha_{i5} = \alpha_{i7} = 2, \alpha_{i3} = \alpha_{i6} = \alpha_{i9} = \alpha_{i11} = \alpha_{i13} = \alpha_{i15} = 1, \alpha_{in} = 0$ for all $i \in \mathcal{I}, n \in \mathcal{T} \setminus \{2, 3, 4, 5, 6, 7, 9, 11, 13, 15\}$ and $\beta_{i2} = \beta_{i3} = \beta_{i4} = \beta_{i5} = \beta_{in} = \beta_{i6} = \beta_{i7} = 1, \beta_{in} = 0$ for all $i \in \mathcal{I}, n \in \mathcal{T} \setminus \{2, 3, 4, 5, 6, 7\}$ where $t_{i1}^* = 3, t_{i2}^* = 4$. The eighth case: $\alpha_{i1} = \alpha_{i2} = \alpha_{i4} = 1$, then $\alpha_{i3} = \alpha_{i5} = \alpha_{i7} = 2, \alpha_{i6} = \alpha_{i9} = \alpha_{i11} = \alpha_{i13} = \alpha_{i15} = 1, \alpha_{in} = 0$ for all $i \in \mathcal{I}, n \in \mathcal{T} \setminus \{1, 2, 3, 4, 6, 5, 7, 9, 11, 13, 15\}$ and $\beta_{i2} = \beta_{i3} = \beta_{i4} = \beta_{i5} = \beta_{i6} = \beta_{i7} = 1, \beta_{in} = 0$ for all $i \in \mathcal{I}, n \in \mathcal{T}$ where $t_{i1}^* = 2, t_{i2}^* = 3, t_{i3}^* = 4$.

Conversely, given a feasible solution $(\alpha, \beta, \mathbf{t}^*)$ to the problem (18), we can construct a feasible solution $\{\mathcal{H}_l\}_{l \in \mathcal{L}}$ for the subset-sum problem by choosing $\mathcal{H}_1 = \{i : 2 \in \Gamma_i^*\}$, $\mathcal{H}_2 = \{i : 3 \in \Gamma_i^*\}$ and $\mathcal{H}_4 = \{i : 4 \in \Gamma_i^*\}$. As $(\alpha, \beta, \mathbf{t}^*)$ is a feasible solution of problem (18), we satisfy the condition $\sum_{l \in \mathcal{L}} \sum_{i \in \mathcal{H}_l} w'_{il} = W'$. \square

A.5. Proof of Proposition 4

Proposition 4 *Let the state variables be binary with $x_{in} \in \{0, 1\}$ for all $i \in \mathcal{I}, n \in \mathcal{T}$. Then, the integrality conditions on the revision decision variables r_{it} for all $i \in \mathcal{I}$ and $t \in \mathcal{P}$ can be relaxed within (AMS_μ) .*

Proof: For any given node pair (m, n) from set $S_{i'} \cap \mathcal{T}(l)$ where $l \in S_i, t' = t + 1, t \in \mathcal{P}, i \in \mathcal{I}$, the NACs can be written as follows:

$$x_{im} \geq x_{in} - (r_{it'} - r_{it}) \quad (19a)$$

$$x_{im} \leq x_{in} + (r_{it'} - r_{it}) \quad (19b)$$

where $r_{it'} - r_{it} \leq 1$ and $r_{i1} = 0$. We distinguish two cases for the values of x_{im} and x_{in} .

Case 1: If $x_{im} \neq x_{in}$ for at least one (m, n) combination, then $x_{im} = 0$ and $x_{in} = 1$ or $x_{im} = 1$ and $x_{in} = 0$. By construction, it is enforced that $r_{it'} = r_{it} + 1$.

Case 2: If $x_{im} = x_{in}$ for all (m, n) , then $x_{im} = x_{in} = 0$ or $x_{im} = x_{in} = 1$. Then, we can consider the following cases: i) $r_{it'} = r_{it}$, ii) $r_{it'} - r_{it} = 1$, and iii) $0 < r_{it'} - r_{it} < 1$. The potential problematic case, which is case iii, implies that at most $\mu - 1$ revisions can be made. This case can only happen if $z(AMS_{\mu-1}) = z(AMS_\mu)$, thus making $\mu - 1$ revisions optimal. In that case, it is still possible to construct an optimal solution with integer revision variables.

These cases show that one can obtain an optimal solution by only considering one by one increase for revision variables in consecutive stages, allowing elimination of the need to enforce integrality of revision variables. \square

A.6. Proof of Proposition 5

Proposition 5 Let $t \in \mathcal{P}$, $t' \geq t$, and $l \in S_t$ be given. Let $K = |S_{t'} \cap \mathcal{T}(l)|$. Without loss of generality, we can order the nodes in $S_{t'} \cap \mathcal{T}(l)$ as $(n, n+1, \dots, n+K-1)$. Then, it is sufficient to include constraints (4b) and (4c) only for the consecutive node pairs (e.g., $(n, n+1), (n+1, n+2), \dots, (n+K-1, n)$) from the set $S_{t'} \cap \mathcal{T}(l)$ as follows:

$$x_{in} \geq x_{i(n+1)} - \bar{x}_i (r_{it'} - r_{it}) \quad n \in S_{t'} \cap \mathcal{T}(l), \quad l \in S_t, \quad t' \geq t, \quad t \in \mathcal{P}, \quad i \in \mathcal{I} \quad (20)$$

We note that we should remove the ordered node $n+K$ from the set $S_{t'} \cap \mathcal{T}(l)$ and add the constraint between $x_{i(n+K-1)}$ and x_{in} , but to enhance readability we slightly abuse notation in constraints (5) by using node $n+K$ for node n .

Proof: We prove this by conditioning on the value of $(r_{it'} - r_{it}) \in \mathbb{Z}_+$.

Case 1: $(r_{it'} - r_{it}) \geq 1$. Then both (4b) and (4c) are nonbinding, as are (5).

Case 2: $(r_{it'} - r_{it}) = 0$. The constraints in (5) will reduce to $x_{in} = x_{i(n+1)} = \dots = x_{i(n+K-1)}$, where $K = |S_{t'} \cap \mathcal{T}(l)|$. Hence, for $m, n \in S_{t'} \cap \mathcal{T}(l)$ as defined in (4b) and (4c), there exists unique $k, k' \in \{0, 1, \dots, K-1\}$ such that $x_{im} = x_{i(n+k)}$ and $x_{in} = x_{i(n+k')}$. Thus, constraints (5) impose $x_{im} = x_{in}$ for all $m, n \in S_{t'} \cap \mathcal{T}(l)$ as defined in (4b) and (4c). \square

A.7. Proof of Proposition 6

Proposition 6 Let $t_{a(m,n)}$ be the stage of the last common ancestor of node pair (m, n) from the set S_t of stage t . It is sufficient to include NACs of (m, n) only for $t, t_{a(m,n)}$ combination, and accordingly, constraints (4b) and (4c) can be replaced by the following constraints:

$$x_{im} \geq x_{in} - \bar{x}_i \left(r_{it} - r_{it_{a(m,n)}} \right) \quad \forall m, n \in S_t, t \in \mathcal{P}, i \in \mathcal{I} \quad (21a)$$

$$x_{im} \leq x_{in} + \bar{x}_i \left(r_{it} - r_{it_{a(m,n)}} \right) \quad \forall m, n \in S_t, t \in \mathcal{P}, i \in \mathcal{I} \quad (21b)$$

Proof: For a given i and node pair (m, n) from set $S_{t'}$, we have NACs (4b) and (4c) for some set of t , where $t \leq t'$, if $(m, n) \in \bigcup_{l \in S_t} S_{t'} \cap \mathcal{T}(l)$. Accordingly, we have NACs for t and $t-v$ where $t = t_{a(m,n)}$ and $v \in \{1, \dots, t-1\}$ as follows:

$$x_{im} \geq x_{in} - \bar{x}_i (r_{it'} - r_{it}) \quad (22a)$$

$$x_{im} \leq x_{in} + \bar{x}_i (r_{it'} - r_{it}) \quad (22b)$$

$$x_{im} \geq x_{in} - \bar{x}_i (r_{it'} - r_{i(t-v)}) \quad v \in \{1, \dots, t-1\} \quad (22c)$$

$$x_{im} \leq x_{in} + \bar{x}_i (r_{it'} - r_{i(t-v)}) \quad v \in \{1, \dots, t-1\} \quad (22d)$$

We can establish the relationship: $r_{i(t-v)} \leq r_{it} \leq r_{it'}$, where $t-v \leq t \leq t'$, based on constraint (4d). This relationship implies that $(r_{it'} - r_{it}) \leq (r_{it'} - r_{i(t-v)})$. Since $(r_{it'} - r_{it})$ attains the lower value, constraints (22a)–(22b) dominate constraints (22c)–(22d) for any $v \in \{1, \dots, t-1\}$, which shows that inclusion of only constraints (22a)–(22b) is sufficient for non-anticipativity. As a result, we conclude that for a node pair (m, n) from set $S_{t'}$, it is sufficient to include NACs only for $(t', t_{a(m,n)})$ combination. For notational brevity, we omit the prime symbol and consider $(t, t_{a(m,n)})$ combination in constraints (21). \square

A.8. Proof of Proposition 7

Proposition 7 Let $t \in \{1, \dots, T\}$ and $t' \geq t$. For given μ , NACs between a particular (t', t) combination become redundant, if $t' > t + (T - \mu) - 1$. Then, constraints (4b) and (4c) reduce to:

$$x_{im} \geq x_{in} - \bar{x}_i(r_{it'} - r_{it}) \quad \forall m, n \in S_{t'} \cap \mathcal{T}(l), l \in S_t, \min(t + (T - \mu) - 1, T) \geq t' \geq t, t \in \mathcal{P}, i \in \mathcal{I} \quad (23a)$$

$$x_{im} \leq x_{in} + \bar{x}_i(r_{it'} - r_{it}) \quad \forall m, n \in S_{t'} \cap \mathcal{T}(l), l \in S_t, \min(t + (T - \mu) - 1, T) \geq t' \geq t, t \in \mathcal{P}, i \in \mathcal{I} \quad (23b)$$

Proof: From Lemma 1, at an optimal solution of (AMS_μ) , $r_{iT}^* = \mu$. Combining this observation with constraints (4d) - (4g) implies that r_{iT} can reach μ only with a sequential increase of revision variables one by one. Consequently, if the difference between t' and t is greater than $T - \mu - 1$, then the difference between $r_{it'}$ and r_{it} is positive. Proof of Proposition 5 shows redundancy of NACs if $r_{it'} - r_{it} > 0$, therefore constraints (23a) and (23b) are sufficient for non-anticipativity. \square

Appendix B: Scenario Tree Generation Algorithm

Algorithm 2 Scenario Tree Generation for Generation Expansion Planning Problem

- 1: Calculate the demand values at the root node as $\{d_{1k} : d_{1k} = w_k d_1\}_{k \in \mathcal{K}}$
 - 2: Obtain capacity factors at root node as $\{l_{i1k}\}_{i \in \mathcal{I}, k \in \mathcal{K}}$ from $\mathcal{N}(\mu_i^l, \sigma_i^l)$
 - 3: **for all** $t \in \{2, \dots, T\}$ **do**
 - 4: **for all** $n \in S_t$ **do**
 - 5: **for all** $k \in \mathcal{K}$ **do**
 - 6: Generate growth factors $\lambda_{nk}^d \sim \mathcal{N}(\mu^d, \sigma^d)$
 - 7: $d_{nk} \leftarrow (1 + \lambda_{nk}^d) d_{a(n),k}$
 - 8: **for all** $i \in \mathcal{I}$ **do**
 - 9: $l_{ink} \sim \mathcal{N}(\mu_i^l, \sigma_i^l)$
-